

DOCTOR EFFICIENCY PREDICTION FOR V-PULSE APPLICATION

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

MADINENI MANIKANTA (Reg. No.37110417)

VUKKURTHI LEELA VENKATSAI (Reg.No.37110404)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by
NAAC JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI - 600 119

MARCH-2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING BONAFI

DE CERTIFICATE

This is to certify that this project report is the bonafide work of **MADINENI MANIKANTA (Reg.No.37110417)** and **V.LEELA VENKATSAI (Reg.No. 37110404)** who carried out the project entitled "**DOCTOR EFFICIENCY PREDICTION FOR V-PULSE APPLICATION**" under my supervision from **November 2020 to April 2021**.

Internal Guide

Dr.A. CHRISTY, MCA, Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Dr. L. LAKSHMANAN M.E., Ph.D.,

Submitted for Vivavoce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

IMADINENI MANIKANTA hereby declare that the Project Report entitled “**DOCTOR EFFICIENCY PREDICTION FOR V-PULSE APPLICATION**” is done by me under the guidance of **Dr.A.CHRISTY MCA, Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. CHRISTY, MCA, Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

World is fully comprised of data. Now a days, data and information rule the world. Data analytics plays a major role in every sector. Predictive analytics a part of data analytics plays a vital role in different sectors of Medicine, Defence, Agriculture, Stock Exchange and many more. Machine learning algorithms are widely used to solve many of these problems. Medicine is essential for every Individual. Patient tracking and disease diagnosis is more important these days. One of the major problem is recommendation of the doctors to patients according to Hospital Management System (HMS). Machine Learning algorithms are trained to solve this problem with higher accuracy. This not only helps Management but also Patients. Python Environment is used for Machine Learning Model whereas Django is used for User Interface(UI) and SQLite for backend Database.

Keywords: Doctor, Django, Hospital Management System(HMS),

Python, SQLite, User Interface(UI).

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	
	LIST OF FIGURES	
1.	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 PROJECT DOMAIN	1
	1.3 METHODOLOGY	2
2.	LITERATURE SURVEY	3
3.	AIM AND SCOPE OF THE PRESENT INVESTIGATION	5
	3.1 AIM OF PROJECT	6
	3.2 SCOPE OF PROJECT	6
	3.3 EXISTING SYSTEM	6
	3.3.1 DISADVANTAGES	6
	3.4 PROPOSED SYSTEM	7
	3.5 FEASIBILITY STUDY	7
	3.5.1 ECONOMIC FEASIBILITY	7
	3.5.2 TECHNICAL FEASIBILITY	7
	3.5.3 SOCIAL FEASIBILITY	8
	3.6 SYSTEM SPECIFICATION	8
	3.6.1 HARDWARE SPECIFICATION	8
	3.6.2 SOFTWARE SPECIFICATION	9
	3.6.3 STANDARDS AND POLICIES	9
4.	EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED	10
	4.1 SYSTEM ARCHITECTURE	10
	4.2 INPUT AND OUTPUT	12

	4.2.1 INPUT DESIGN	12
	4.2.2 OUTPUT DESIGN	15
	4.3 TESTING	15
	4.4 DESIGN PHASE	21
	4.4.1 DATA FLOW DIAGRAM	21
	4.4.2 UML DIAGRAMS	22
	4.4.3 FLOW CHART	23
	4.4.4 SEQUENCE DIAGRAM	24
5.	RESULTS AND DISCUSSION	26
6.	CONCLUSION AND FUTURE WORK	31
	REFERENCES	32
	APPENDIX	
	A. JOURNAL PAPER	34
	B. PLAGIARISM REPORT	46
	C. SOURCE CODE	49

LISTOFFIGURES

FIGURENo.	FIGURENAME	PAGENO.
4.1	ARCHITECTURE DIAGRAM	10
4.2.1	DATA BASE	11
4.2.2	MICROSOFT VISUAL STUDIO	16
4.4.1	DATA FLOW DIAGRAM	21
4.4.2	UML DIAGRAM	22
4.4.3	FLOW CHART	23
4.4.4	SEQUENCE DIAGRAM	24
5.1	EFFICIENCY	26
5.2	FIND DOCTOR	27
5.3	STATISTICS	28
5.4	PHARMACY	29
5.5	ADMIN PAGE	30

CHAPTER1INTROD UCTION

INTRODUCTION

Medicinal field is one of the important and emerging sector in today's world. Hospitals play a vital role in saving many lives and increasing medicinal values. Due to the mechanical life of today's individual drastically change is observed in health issues when compared to past. For hospitals the management of patients with upcoming health issues is huge task to be handled effectively. Hospital Management System is the Software approach used by majority in medical field for maintaining Patient records, Pharmaceutical Records, Doctor Records and many more. V-Pulse is an efficient application developed by VNC Digital Services Pvt LTD for HMS on Spring Boot Platform of JAVA. It is used by many hospitals for the records purpose. It not only works as an Database platform but also an efficient application for Statistical analysis of Data. HMS makes the management's work easier for maintenance of hospital data. Doctor efficiency is one of the important aspect to be calculated for hospital management to consider doctor for future reference because in a technological world of today people are choosing based upon reviews and successful cases solved by particular doctor or hospital management. So, It is necessary to predict the efficiency of doctor and rank them upon number of cases solved and complexity of each case. The Disease level complexity is defined by the previous study of diseases and references by many of doctors in all aspects.

1.2 PROJECT DOMAIN

To solve a real-time problem on a computer, we would like an algorithm. An algorithm is a sequence of instructions that should be carried out to transform the raw data input to required output. For example, one can devise an algorithm for sorting. The input may be a set of numbers and therefore the output is their ordered list. For an equivalent task, there could also be various algorithms and that we could also be curious about finding the foremost efficient one, requiring the smallest amount number of instructions or memory or both. Machine Learning is the way of training the algorithms with different types of data to

get more accuracy in the resultant output. In this Project we have used both Supervised Machine Learning techniques in order to get a clear and efficient output. Supervised learning makes training with known output variables with two prominent techniques namely classification and regression. It is study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of AI. Machine learning algorithms build a mathematical model supported sample data, mentioned as "training data", so on form predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are utilized in a good sort of applications, like email filtering and computer vision, where it's difficult or infeasible to develop a standard algorithm for effectively performing the task.

1.3 METHODOLOGY

The Main logic of code for prediction runs in spyder console of anaconda environment written in python language at the backend. Data base acts as middleware where all the data is stored for the process. Database used here is My SQL workbench and SQLite3 as depends upon the system configurations. The User Interface is developed using Django frame work of python in visual studio and UI screen appears in internet explorer using the static uniform resource locator of the Django frame work. The data fed happens through the admin page for the patient record. The data is stored in database and retrieved in ML source code of anaconda frame work. The training and testing of the data happens and validation is checked. The validated and predicted outputs are directly exported to the database. The results are retrieved and displayed in the User Interface Screen according to the user input and specification.

CHAPTER 2

LITERATURE SURVEY

The author[1] in this paper mainly discusses about smart Hospital Management System (SHS). It is an infrastructure component relies heavily on the actual resources made available to it for its proper functioning, operation and maintenance. This is a solution aimed to present architecture Integration Framework using TOGAF's Architecture Development Method. The TOGAF's Architecture consists mainly of 8 phases namely Architecture Vision, Business Architecture, Information Systems Architecture, Technology Architecture, Opportunities and Solutions, Migration Planning, Implementation Governance, Architecture change Management. The effectiveness of SHS is explained in terms of TOGAF with advantages and disadvantages.

The author[2] discusses about the benefits and problems that can be seen with the use of electronic versions of medical records. They constitute the permanent documentation of patient health, permitting the medical professional to evaluate symptoms and signs within a broader temporal perspective, contributing to improvements in making diagnoses and providing treatment. The hospital information systems constitute of three levels namely Operational level, Tactical level and Strategic level about three operational areas of Medical area, Administrative area and Hospitality. It also discusses about case study of testing made on HMS of three hospitals in city of Sao Paulo, Brazil which includes~ visualization and testing results of HMS.

This author[3] in this discusses about an effective mechanism for chronic disease risk prediction by mining the data containing historical health records and personal life style information. The method outperformed is the traditional mechanism in terms of accuracy, precision and sensitivity for predicting the risk of diabetes. In particular, insightful observations show that the consideration of life-style information can effectively enhance whole performance for risk prediction. The framework described here is divided into three phases. In the first phase, health risk patterns of each item are found from a sequence dataset of each item in a health examination and a lifestyle datasets respectively. The second phase, each health risk pattern is as a feature attribute and constructed a dynamic feature datasets for both Static Dynamic data respectively. The health risk prediction model is deployed using classification algorithm. Finally, in the third phase the

patient's data to be predicted, the examinee's historical data related to health and lifestyle is converted to the suitable dynamic and static data according to the health risk patterns.

The author[4] explains about efficient patient management system based on Near Field Communication (NFC) technique. For registration each NFC tag/ wrist band is identified by a Unique Identification Number (UIN) that can be programmed and can also be protected through password. NFC encoded device can be used to read this information from the patient at any time without causing any inconvenience to them. NFC is mainly introduced to reduce cost and paper work for Hospital Management System (HMS). The system of the NFC based Hospital management system consists of six layers namely physical layer, middle layer, process layer, data access layer, application layer and user interface layer.

In this paper author[5] explains about the Machine learning algorithms are used for effective prediction of chronic disease outbreak in disease-frequent communities. Latent factor model was used to reconstruct the missing data. Convolutional Neural Network-CNN based multimodal disease risk prediction algorithm is used for structured and unstructured data from hospital. The prediction accuracy was about 98.4 when compared to other models of Big data analytics. The Data used was real-time hospital data stored in cloud from years of 2013-2015 which includes 31919 patients of 20320848 records on the whole. Stochastic Gradient Descent algorithm and CNN-based UNI model disease risk prediction algorithm are used for Structured and Unstructured data respectively.

This author[6] speaks about the Artificial Intelligence (AI) based health physician system that would be able to interact with the patient, do the diagnosis and suggest quick remedy or treatment of their problem. The system uses a questionnaire based approach to query the user (patient) about various Symptoms, based on which a decision is made and a medicine is recommended. The usage of AI is because it make systems behave and work more like humans is gaining popularity. Natural Language Processing (NLP), smart agents, Machine Learning (ML) and knowledge base were used to make Rapid Application model for Smart Doctor. Decision Tree Algorithm is used here for every disease for the medicine prediction and doctor recommendation. This Software acts as efficient chatbot for the patient convenience.

The author[7] discusses about the importance of solutions for improving public health, healthcare providers are required to be fully equipped with appropriate infrastructure to systematically generate and analyze big data, whereas it need an efficient management, analysis, and interpretation. As the data is larger in size is not said to be big data because it should satisfy the conditions of velocity, veracity, volume and variety. The collected data is stored in data warehouse and workflow is progressed by analytics 4.0 namely descriptive analytics(1.0), Diagnostic analytics(2.0), Predictive Analytics(3.0) and prescriptive analytics(4.0) for future usage. The healthcare information mainly consist the of records mainly of Electronic health record, clinical information and patient information. This brought a drastic change in journals publishing in health care associated with big data from 2000 to 2018. The IBM Watson is used here for Natural Language processing and Deep learning process.

In this paper author[8] explains about impact of usage of Big Data analytics in health care. One of the objective in this paper is to address healthcare delivery problems and improve healthcare quality because Pharmaceutical-industry experts and shareholders have begun to routinely analyze big data to obtain insight, but these activities are still in the early stages. Predictive analytics plays a vital role in this analytics where it has been recognized as one of the major business intelligence approaches, but its real world applications extend far beyond the business context. It not includes text analytics but also multimedia analytics whereas Hadoop architecture is used for health informatics.

CHAPTER 3

PROJECT DESCRIPTION

3.1 AIM OF PROJECT

The project is designed for the benefit of hospital management. The objective is to find the efficiency of a doctor based on the number of solved successful cases. It is easy to predict the success rate of a doctor using number of cases he/she deals with but success rate is not the important thing in today's world. The doctor solved cases is taken with the complexity of each case he solves. This approach not only involves doctor's success rate but also the complex rate of disease. This prediction is made through Machine Learning algorithms easier for the problem to be solved. The analytics 4.0 is used where descriptive and diagnostic analytics is used for the disease level complexity prediction by Decision trees and Predictive and Perspective analytics are used for Doctor Efficiency prediction.

3.2 SCOPE OF THE PROJECT

This project has future scope of deploying in server based mechanism for better results. Treatment level complexity is one more case where different levels of treatment for same disease. The future scope of implementing the complexity levels uses Artificial Neural Networks. This implementation can be made as a qualitative approach as it needs high end discipline to work in. The proposed solution is developed depicts the outputs acquired at an higher rate of accuracy.

3.3 Existing System

Hospital Management System (HMS) is an effective system of all records and statistics are stored explicitly. In this corporate world many IT companies have created many HMS applications which are being used by many hospitals for storing and analyzing their data explicitly. One among the applications is V-Pulse which is created by VNC Digital Services Pvt LTD used by many hospitals has many features like patient, doctor record maintenance, pharmaceutical data analysis with many other features. Other systems have many other features like medicine recommendation system, patient treatment reminder system and many more. Many applications have no module or model of doctor recommendation system for management system or patient purpose.

3.3.1 Disadvantages:

1. Many other systems do not have prediction system or recommendation of doctors.
2. Doctor Efficiency system based on success and failure cases calculation is not feasible.
3. According to Literature survey many reliable approaches has not given better solution to this doctor efficiency prediction system.
4. Doctor recommendation system for particular disease with complexity of disease is no where mentioned and developed.

3.4 Proposed System

The proposed system is an application which recommends doctor for particular disease with purpose number of successful cases solved and efficiency too. This also explore about the top five doctors of particular specialization. This also has Pharmaceutical analysis according to the department wise of the concerned hospital. The HMS of our system named as V-Pulse which was developed under Java platform of Springboot for backend and Angular for front end by VNC Digital Services Pvt LTD. The recommendation system was developed under Python Language. Machine Learning Algorithms are developed using Anaconda environment in Spyder IDE. This is connected as backend system to middleware database. The Database is either of MySQL workbench and SQLite which depends upon the server configuration used. The front end is the main User Interface which was developed under Django framework in Visual Studio.

3.5 Feasibility Study

The feasibility study of a project is to study about the project which fulfils the needs of today's market stage by considering different complications in development of the project. This feasibility analysis of project depicts a clear view to make project as a business plan as it need to be commissioned. This is to guarantee that the proposed framework isn't a weight to the organization.

Two key considerations involved in the feasibility analysis are:

1. Economic Feasibility.
2. Technical Feasibility.
3. Social Feasibility.

3.5.1 Economic Feasibility

This is study of cost of the project is to be low for the development so that do not the economic

situation of the organisation where it is being developed. This project is developed using the software's which are at free of cost. The data is deployed from the company's V-Pulse cloud server. The Economic feasibility is higher for the proposed system.

3.5.2 Technical Feasibility

This is a study of the technological aspects of the project which is acceptable in this era of technological world. This project is developed using the Machine Learning Strategies in python environment which is widely acceptable. The domain of the project is a very much feasible and acceptable widely as it is a trending in this world.

3.5.3 Social Feasibility

This is a study of the socialistic aspects of the project. This explains about the User experience about the project. The proposed system is a user friendly system where the user screens are feasible and easily interpretable. This project has a high range of social feasibility and effective user friendly experience.

3.6 System Specification

3.6.1 Hardware Specification

- Processor - Core i5 or i7 @1.70GHz 2.40GHz
- RAM - 8 GB
- Hard Disk - 500 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two Button Mouse

3.6.2 Software Specification

- Operating System - Windows 7,8,10
- ML source code environment - Python using Anaconda prompt
- Data Base - My SQL workbench or SQLite3 browser
- User Interface code - Microsoft Visual Studio for Django

3.6.3 Standards and Policies

Anaconda Prompt

It is a type of command line interface which explicitly deals with the Machine Learning modules. The navigator is available in Windows, Linux and MacOS. This has a huge number of IDE's which makes us easier in coding. The user Interface also can be implemented in python.

Standard Used: ISO/IEC 27001

SQLite Database

It is data base prompt which is used to implement all SQL commands. This is used as an middleware system which contacts between the Machine Learning source code and User Interface screens. This can be integrated easily with python IDE and Django interface.

Standard Used: ISO 8601

Microsoft Visual Studio

The Microsoft Visual Studio version 16.0.12 is used for creating the User Interface. In this Django frame work is implemented. UI screens are designed using html and css where the frame work of Django for

Integration with database and Machine Learning source code.

Chapter 4

EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

4.1 System Architecture

The simpler architecture has been designed of Prediction System. This explains about all the phases of Implementation System.

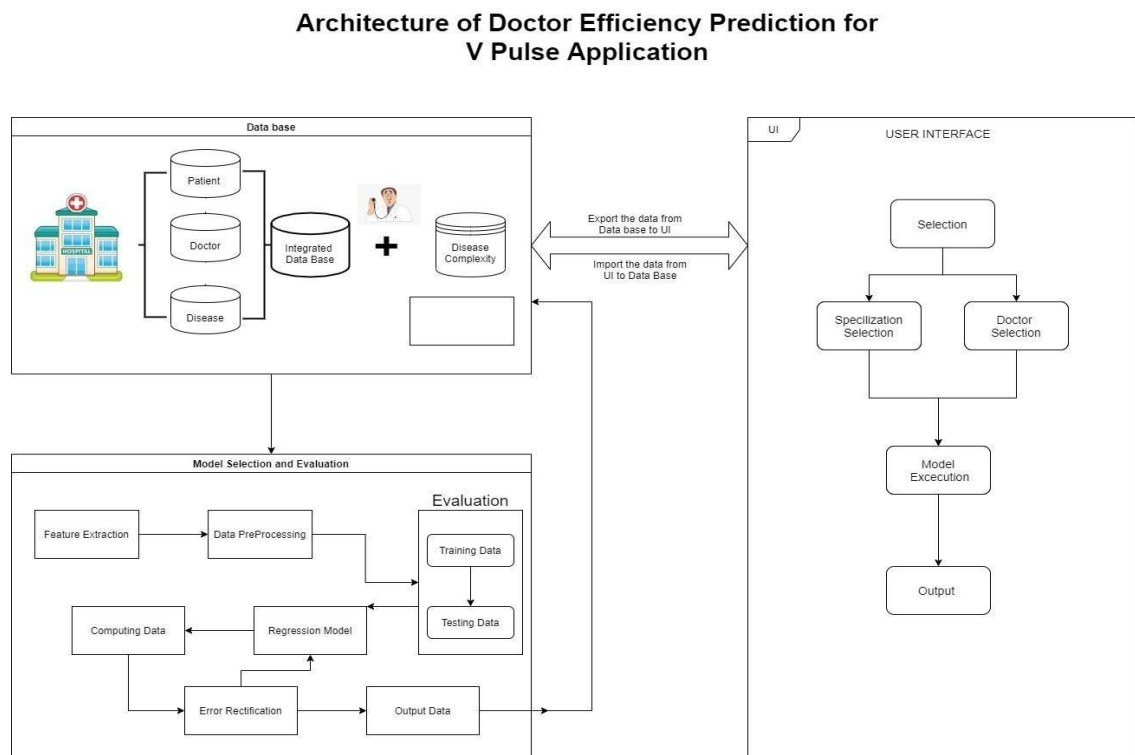


Figure 4.1: Architecture Diagram

The front end is User Interface (UI) which is connected to middleware Database integrated with the backend Machine Learning source code. The database is collective repository of Patient records, Doctor Records and Disease records. These three databases are collectively integrated to a single database with many tables. To this database disease complexity is added from doctor's perception and with the factors defining the disease complexity for new diseases is done through

Decision Tree Algorithm. This module is for hospital management for the Disease Level Complexity (DLC) calculation purpose. The data from integrated database flows into the main code (or) source code.

In this source code the initially data pre-processing happens where the noise data and irrelevant attributes are dropped. The complexity in raw data acts as tuple it is changed to attribute using pre-processing techniques. Feature Selection and Feature Extraction is done for this data for better algorithm induction. This data is splitted into training and testing data and Linear Regression Algorithm is applied. The predicted data from algorithm is compared with that of original data and checked for accuracy. This final data is stored into database.

These final data is been stored into new table in database so that it is retrieved in UI screens. UI screens are created using Django frame work in python language. There are four user screens in our UI. The first screen displays doctor efficiency according to the specialisation selected as one need to select the specialization first and list comes accordingly and when doctor name is selected the efficiency is shown. The second screen displays efficiency of doctors using the name of particular doctor specifically. The third screen acts as ranking system where it ranks the top most five doctors for a specific department. The final screen explores about the pharmaceutical data according to department wise. The admin page is used to check with raw data and adding new data to database. The crud operations in database can be applied from UI admin page itself. The number of users and groups can be checked using Admin Page itself.

Implementation phase is an essential and crucial phase in development of a project. After all the feasibility studies of a project are made and methodologies are studied it is implemented into a project or a working module. The execution includes cautious arranging, examination of the current framework and its requirements on usage, planning of techniques to accomplish changeover and assessment of changeover strategies. All the strategies are taken under similar platforms where modified to better working model.

4.2 Input and Output

4.2.1 Input Design

The input to the model of is different according to system and algorithm performed. In the project two algorithms are implemented.

Disease Name	Cause	Genetic	RepitativeOccurance	Success rate	Complexity
APPENDICITIS	GENERAL	NO	NO	HIGH	ONE
BLEEDING GUMS	GENERAL	NO	YES	HIGH	ONE
BODY PAINS	GENERAL	NO	YES	HIGH	ONE
CAVITIES	BACTERIA	NO	YES	HIGH	ONE
COLD	VIRUS	NO	YES	HIGH	ONE
DRY SKIN	GENERAL	NO	YES	HIGH	ONE
HEAD ACHE	GENERAL	NO	YES	HIGH	ONE
INSOMNIA	GENERAL	NO	YES	HIGH	ONE
OBESITY	GENERAL	NO	YES	HIGH	ONE
RICKETS	GENERAL	YES	NO	HIGH	ONE
ROOT CANAL	GENERAL	NO	YES	HIGH	ONE
SOLAR RETINOPATHY	GENERAL	NO	YES	HIGH	ONE
SPRAINED ANKLE	INJURY	NO	YES	HIGH	ONE
STOMACH ACHE	GENERAL	NO	YES	HIGH	ONE
STRESS	GENERAL	NO	YES	HIGH	ONE

Table 4.2.1: Table for DLC calculation

Firstly data input contains attributes like disease name, genetic, repetitive, success rate and

complexity which contribute to DLC calculation using decision tree algorithm. The second module is to predict the doctor efficiency according to solved number of cases and their complexity. This data table contains attributes of Patient id, Consultant name, Specialization, Disease and Complexity which are used to predict the doctor efficiency using Linear Regression Algorithm.

Patient ID	Consultant	Specialization	Disease	Complexity
1592	Ajay Kumar	Urologist	bladder prolapse	4
1593	Ajay Kumar	Urologist	prostate cancer	5
1594	Ajay Kumar	Urologist	prostatitis	1
1595	Ajay Kumar	Urologist	hematuria	3
1596	Ajay Kumar	Urologist	overactive bladder	2
1597	Ajay Kumar	Urologist	bladder prolapse	4
1598	Ajay Kumar	Urologist	prostatitis	1
1599	Ajay Kumar	Urologist	hematuria	3
1600	Ajay Kumar	Urologist	overactive bladder	2
1601	Ajay Kumar	Urologist	bladder prolapse	4
1602	Ajay Kumar	Urologist	prostatitis	1
1603	Ajay Kumar	Urologist	prostate cancer	5
1604	Ajay Kumar	Urologist	overactive bladder	2
1605	Ajay Kumar	Urologist	prostatitis	1
1445	Ashok L. Kirpalani	Gynecologist	amenorrhea	3
1446	Ashok L.	Gynecologist	menstrual	4

	Kirpalani		disorder	
1447	Ashok L. Kirpalani	Gynecologist	uterine fibroids	2
1448	Ashok L. Kirpalani	Gynecologist	amenorrhea	3
1449	Ashok L. Kirpalani	Gynecologist	pelvic pain	1
1450	Ashok L. Kirpalani	Gynecologist	menstrual disorder	4
1451	Ashok L. Kirpalani	Gynecologist	pelvic pain	1
1452	Ashok L. Kirpalani	Gynecologist	endometriosis	5
1453	Ashok L. Kirpalani	Gynecologist	uterine fibroids	2
1454	Ashok L. Kirpalani	Gynecologist	amenorrhea	3
1455	Ashok L. Kirpalani	Gynecologist	endometriosis	5

Table 4.2.1: Table for Doctor Efficiency calculation

These both tables and output tables are stored in same database stored in DB SQLite3 version. This database is directly integrated with source code and UI code in Django framework.

Database Structure		
Browse Data Edit Pragma Execute SQL		
Create Table Create Index Modify Table Delete Table Print		
Name	Type	Schema
Tables (14)		
app_pharmacydata	CREATE TABLE	"app_pharmacydata" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "Specialization" varchar(300) NOT NULL)
app_rawdata	CREATE TABLE	"app_rawdata" ("Patient_ID" INTEGER, "Consultant" TEXT, "Specialization" TEXT, "Disease" TEXT, "Complexity" REAL,
app_results	CREATE TABLE	"app_results" ("id" REAL, "Consultant" TEXT, "Specialization" TEXT, "Cases" REAL, "Efficiency" REAL)
auth_group	CREATE TABLE	"auth_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE)
auth_group_permissions	CREATE TABLE	"auth_group_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group_id" integer NOT NULL REFERENCES "auth_group" ("id"), "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id"))
auth_permission	CREATE TABLE	"auth_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id"), "codename" varchar(100) NOT NULL UNIQUE)
auth_user	CREATE TABLE	"auth_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" datetime NOT NULL, "is_superuser" boolean NOT NULL, "username" varchar(30) NOT NULL UNIQUE, "first_name" varchar(30) NOT NULL, "last_name" varchar(30) NOT NULL, "email" varchar(254) NOT NULL)
auth_user_groups	CREATE TABLE	"auth_user_groups" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id"), "group_id" integer NOT NULL REFERENCES "auth_group" ("id"))
auth_user_user_permissions	CREATE TABLE	"auth_user_user_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id"), "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id"))
django_admin_log	CREATE TABLE	"django_admin_log" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "action_time" datetime NOT NULL, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id"), "object_id" integer NOT NULL, "action_flag" integer NOT NULL, "change_message" text NOT NULL)
django_content_type	CREATE TABLE	"django_content_type" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app_label" varchar(100) NOT NULL, "model" varchar(100) NOT NULL UNIQUE)
django_migrations	CREATE TABLE	"django_migrations" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app" varchar(255) NOT NULL, "name" varchar(255) NOT NULL, "timestamp" datetime NOT NULL)
django_session	CREATE TABLE	"django_session" ("session_key" varchar(40) NOT NULL PRIMARY KEY, "session_data" text NOT NULL, "expire_date" datetime NOT NULL)
sqlite_sequence	CREATE TABLE	sqlite_sequence(name,seq)
Indices (15)		
auth_group_permissions_group_id_b120c...	CREATE INDEX	"auth_group_permissions_group_id_b120c9f9" ON "auth_group_permissions" ("group_id")
auth_group_permissions_group_id_permis...	CREATE UNIQUE INDEX	"auth_group_permissions_group_id_permission_id_0cd325b0_uniq" ON "auth_group_permissions" ("group_id", "permission_id")
auth_group_permissions_permission_id_8...	CREATE INDEX	"auth_group_permissions_permission_id_84c5c92e" ON "auth_group_permissions" ("permission_id")
auth_permission_content_type_id_2f476e4b	CREATE INDEX	"auth_permission_content_type_id_2f476e4b" ON "auth_permission" ("content_type_id")
auth_permission_content_type_id_codena...	CREATE UNIQUE INDEX	"auth_permission_content_type_id_codename_01ab375a_uniq" ON "auth_permission" ("content_type_id", "codename")
auth_user_groups_group_id_97559544	CREATE INDEX	"auth_user_groups_group_id_97559544" ON "auth_user_groups" ("group_id")
auth_user_groups_user_id_6a12ed8b	CREATE INDEX	"auth_user_groups_user_id_6a12ed8b" ON "auth_user_groups" ("user_id")
auth_user_groups_user_id_group_id_94350...	CREATE UNIQUE INDEX	"auth_user_groups_user_id_group_id_94350c0c_uniq" ON "auth_user_groups" ("user_id", "group_id")
auth_user_user_permissions_permission_i...	CREATE INDEX	"auth_user_user_permissions_permission_id_1fbb5f2c" ON "auth_user_user_permissions" ("permission_id")
auth_user_user_permissions_user_id_a95e...	CREATE INDEX	"auth_user_user_permissions_user_id_a95ead1b" ON "auth_user_user_permissions" ("user_id")
auth_user_user_permissions_user_id_perm...	CREATE UNIQUE INDEX	"auth_user_user_permissions_user_id_permission_id_14a6b632_uniq" ON "auth_user_user_permissions" ("user_id", "permission_id")
django_admin_log_content_type_id_c4bc...	CREATE INDEX	"django_admin_log_content_type_id_c4bce8eb" ON "django_admin_log" ("content_type_id")
django_admin_log_user_id_c564eba6	CREATE INDEX	"django_admin_log_user_id_c564eba6" ON "django_admin_log" ("user_id")
django_content_type_app_label_model_76...	CREATE UNIQUE INDEX	"django_content_type_app_label_model_76bd3d3b_uniq" ON "django_content_type" ("app_label", "model")

Figure 4.2.1: DataBase

4.2.2 Output Design

The output is made through UI screens designed using Django framework in python Language. Microsoft Visual Studio is used to create all UI screens. Html and css languages are used to create pages, borders and colours for them. There are four screens of UI and an Admin Page is present in our module. These screens are integrated with database and source code.

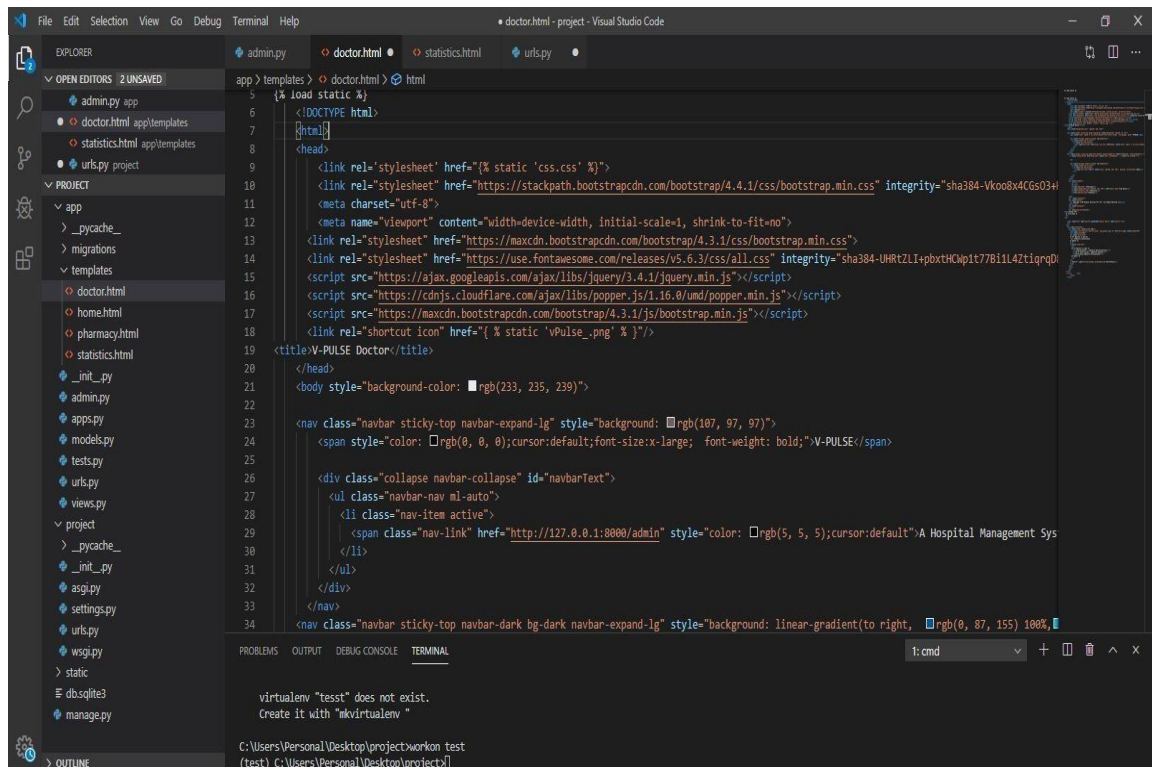


Figure 4.2.2: Microsoft Visual Studio

4.3 Testing

Testing is an integral part of project where errors are rectified using Manual and Automation testing. . It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. There are many types of testing used here for error detection in case of integration of application and deployment. User Interface is also tested here for better experience.

Types of Testing

Unit testing

Source code checking is performed by writing and running test programs against it. The Developers write the test programs according to source code which are executed by testers and are subjected to either pass or fail. This testing is similar to black box testing for Graphical User Interface (GUI) while unit is performed for memory based functioning.

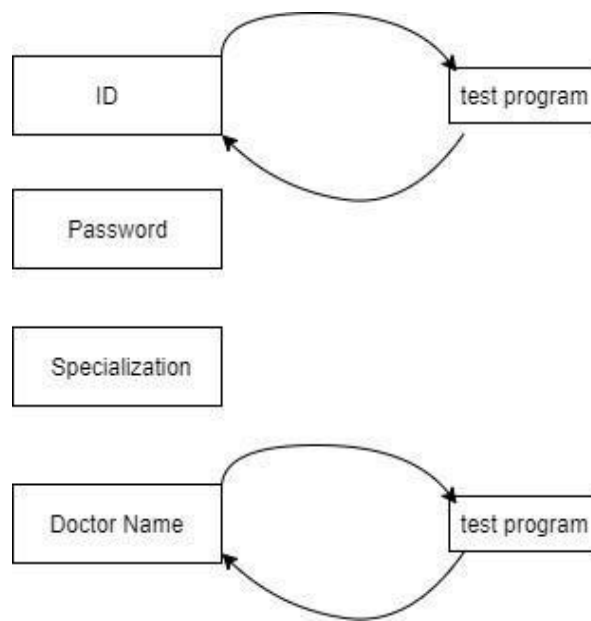


Figure 4.3: Unit testing

Test result

In our Project unit testing is performed for all UI components starting from login page. The ID and password fields are checked here by running tests programs against them and results are checked for space complexity. This applies for all of cases including admin page too.

Integration testing

Software is said to be a collection of modules where developer integrates two or modules as per requirement of client. In our project integration is done between Database, ML model and User Interface. Firstly Integration is done between database and ML source code in Spyder IDE to check the raw data flow into code and output to be stored in database. The other step of integration is done between User Interface and Database when buttons in UI are clicked data flow is checked for accuracy and performance. Among the types incremental integration testing is applied. Sandwich approach is used to test between modules.

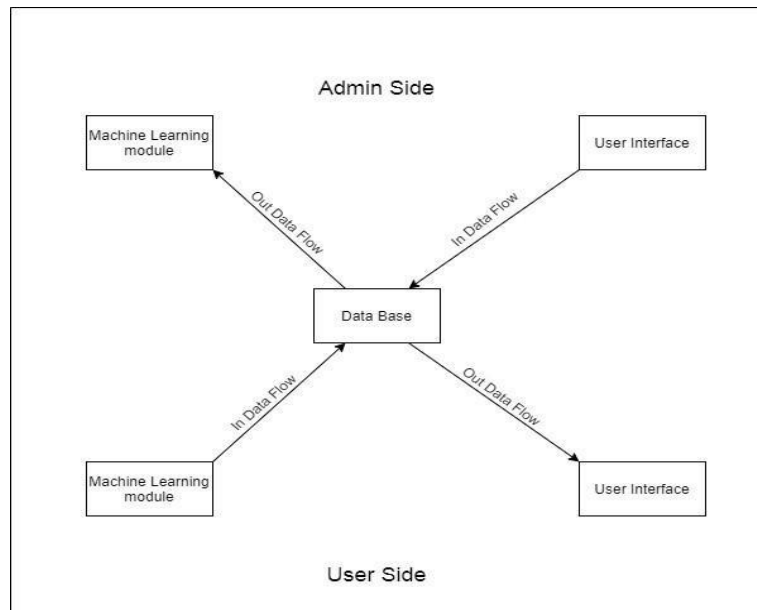


Figure 4.3.1: Integration testing

Test result

Parent module is Data base where input data for admin and retrieve data for efficiency is done through User Interface which is one child class and after data is in database the flow is from Database to ML module for prediction. ML module is one more child class. Sandwich testing is applied to this clock cycle to check data flow in all these modules. This test result has shown higher accuracy as there is no data lag, redundancy and loss of data in any of these modules. The Sandwich approach diagram depicts clearly about testing approach.

Functional testing

This testing is applied to URL components and functional components where each component of URL is tested. Firstly URL domain is tested for security as of `https://` format. If it is not secure data breach may happen. After security pre condition statement is checked namely for visibility and name recommendation. URL components may be static or dynamic based on node of creation but here in our case of prediction it is a static component developed using Django frame work where "`http://127.0.0.1:8000/app/`" is created as a static component. After URL components are tested functional components are tested for their functionality performance. Initially functions are needed to be identified and create input data based on functions as test cases are needed to be created for checking accuracy of test case outputs. Comparing test case scenarios with original output makes a clear note of test accuracy and functional accuracy.

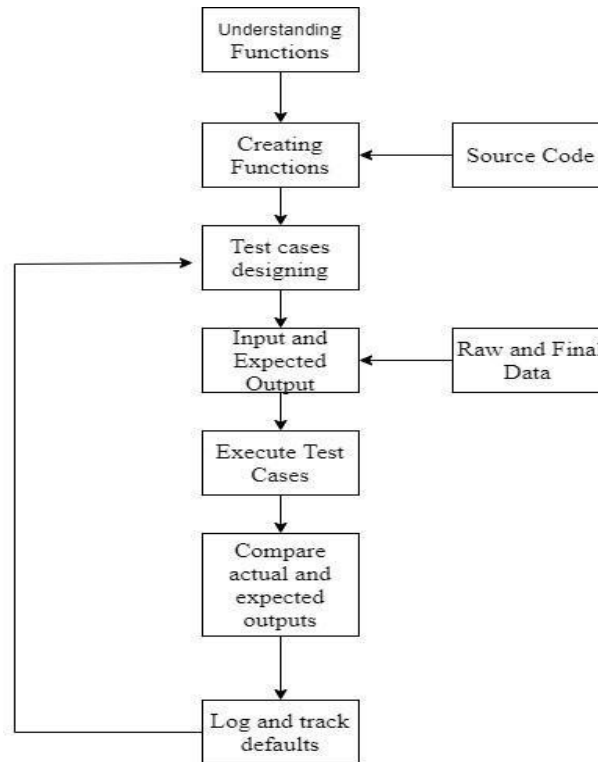


Figure 4.3.2: Functional testing

Test Result

In our efficiency prediction system functional components are created for source code as of pre-process data and feature extraction data as one function and algorithmic data as another function. Test cases are created for both scenarios and functional components are tested. As pre-process data has a noticeable functional error test case failed in case of that scenario.

White Box Testing

This is also known as Open box testing or Glass Box testing. In this is type of testing where we check the source code which was written for software development. It is performed by the developers. Here in case of our Doctor Efficiency Prediction system the source code is converted in flow chart and test cases are performed for output analysis. For Doctor Efficiency Prediction code does not contain any of loops but only condition here is to check whether doctor name with accuracy present in data for input or not. Path testing here checked only while integrating ML model with Database connection as raw data and output data tables belong to same database. If error occurs database gets locked and Data redundancy occurs. Statement testing is performed for all statements present in flow chart. It checks all executable control statements and Initializing statements. There are no possibilities of conditional statements in any of our logic related in Decision tree or Regression in source code. So, statement testing is done for input statements in

pre-processing and ML Stage. Path testing is performed for source code where all control flow of code executes once without interrupts and errors. In this System the path as follows from raw data frame to pre-processing code where duplicates are removed and complexity and efficiency are added into a new data frame followed by ML algorithm for prediction, as predicted outputs. Condition testing is performed to check for basic condition and compound condition where all conditions should execute for true or false value's. In our solution for Doctor Efficiency the only condition is the data driven in to algorithm should have an output latent variable as it is supervised learning. Loop testing can also be performed for source code but no loops exist in our code for Doctor Efficiency and Disease complexity.

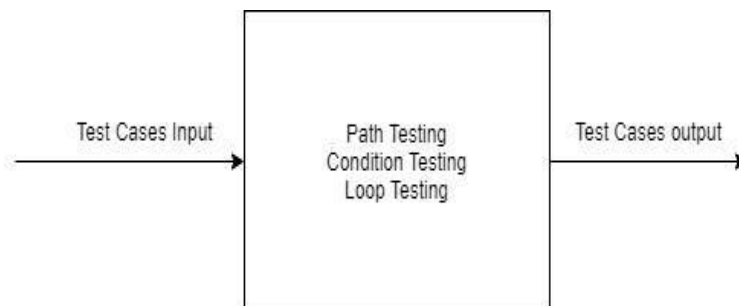


Figure 4.3.3: White Box Testing

Black Box Testing

This is also known as Non-Structural testing or closed box testing. This not only used for checking Graphical User Interface (GUI) of the System but also Behaviour of software. This is performed by test engineer. Test scenarios are written where all possible ways of testing the Software are mentioned. Templates of test scenarios contain Precondition, procedure and expected output.

The testing scenario starts with pre-condition where the characters of Uniform Resource Locator (URL) in address bar are not only checked for visibility but also editable format. After typing initial letters suggestions are checked correctly like our system which was developed in local host, server name is automatically created by "Django" frame work. Suggestions are not considered in this case.

The procedure follows with check of font size, colour, alignment satisfies requirement completely. This also includes inferring negative scenarios of UI testing. System testing is also performed using the black box testing itself. In this type end to end testing is done from Doctor Efficiency prediction using specialization selection to Pharmaceutical statistics. This involves test of four modules of User interface leaving Admin Page. Admin page is tested separately.

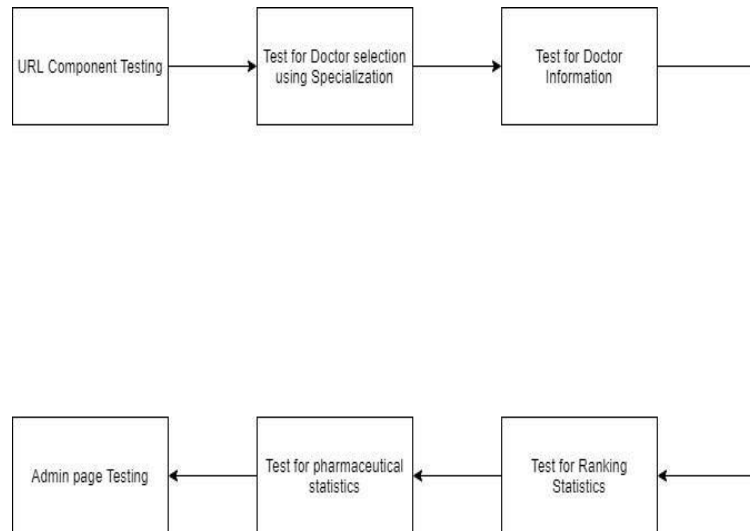


Figure 4.3.4: Black Box Testing

4.4 Design Phase

4.4.1 Data Flow Diagram

The DFD depicts about the flow of data in the prediction system. This flow happens between all the modules of the prediction system.

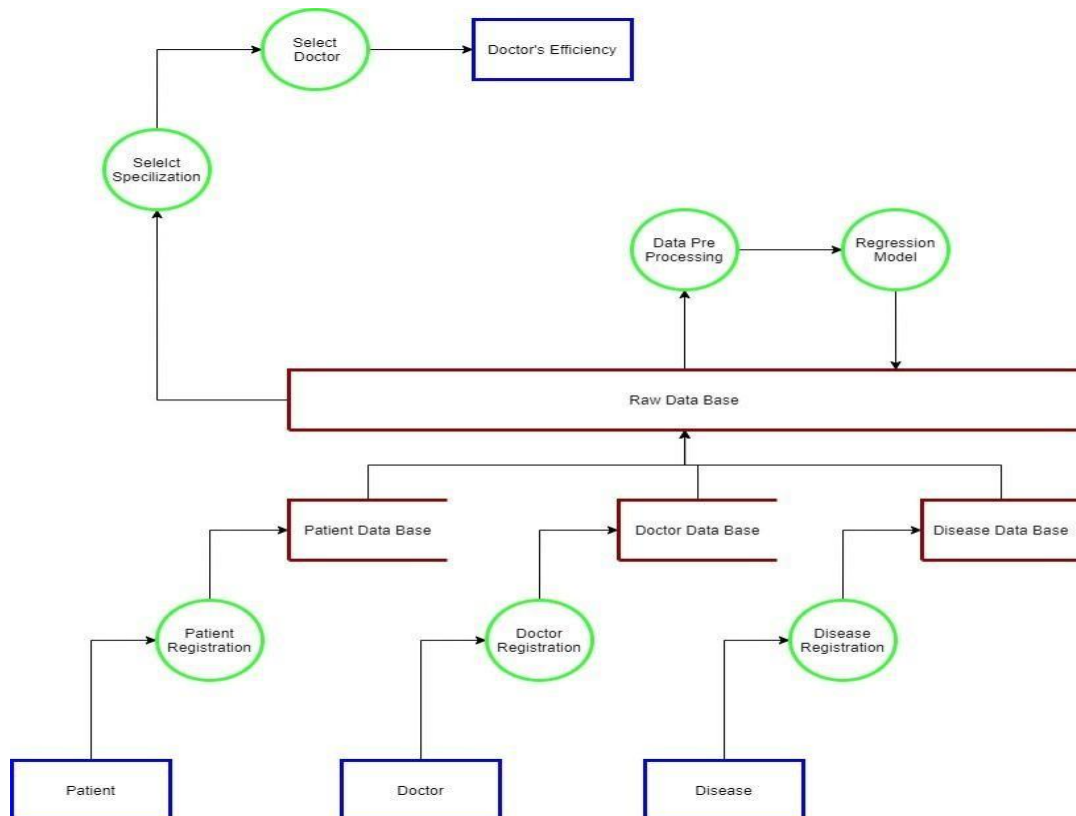


Figure 4.4.1: Data Flow Diagram

Primary data flow happens between the data flow of registrations. The patient details, doctor details are mainly collected and stored in database. The disease registration happens only if particular disease is not present in database. The DLC classification happen using Decision Tree Algorithm. These databases are integrated to raw database for algorithm prediction purpose.

The predicted data flow next flows back to the same database and stored in new table. This contains values of predicted efficiencies of doctors. The next flow happens from database to UI screens where predicted results are displayed according to selection of management in all of four screens. The data entered in admin page flows in form of registrations separately in different tables and next stored in database. This data flow happens in transparent manner where no data loss is being happened.

4.4.2 UML Diagram

Unified Modelling Language diagram is used to depict the whole scenario of model. The prediction model has different data interactions and explains them on the whole.

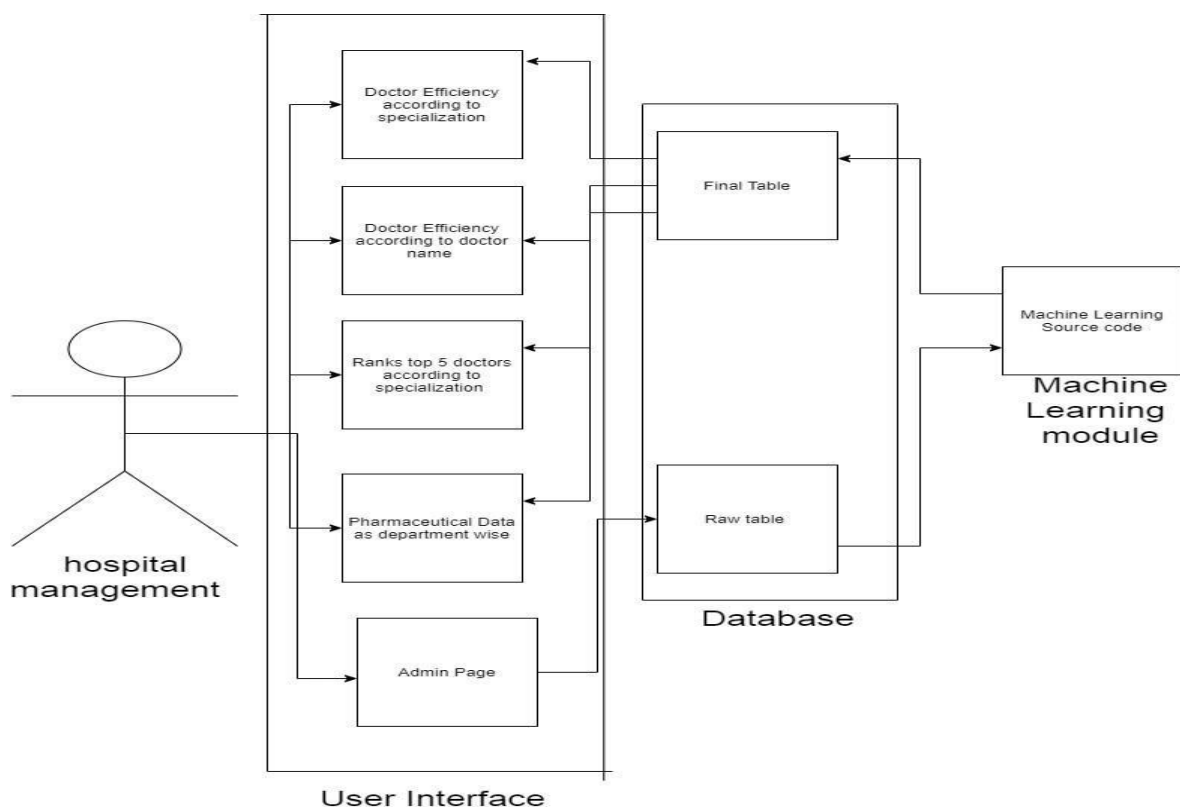


Figure 4.4.2: UML Diagram

The hospital management is the end user in our prediction system. They access directly to our application through web server and details can be checked through UI screens according to their particular usage. As mentioned in prior our system has five feasible UI screens which are easily

accessible and used. Data from admin page is directly driven to raw table and next to source code.

In source code the prediction system is compiled efficiently and used for efficiency prediction. The predicted efficiency data is stored in final table. Raw table and final table are present in same database. This data is fetched into all screens of UI as per user requirement.

Database used here is My SQL workbench and SQLite3 as depends upon the system configurations. The User Interface is developed using Django frame work of python in visual studio and UI screen appears in internet explorer using the static uniform resource locator of the Django frame work. The data fed happens through the admin page for the patient record. The data is stored in database and retrieved in ML source code of anaconda frame work. The training and testing of the data happens and validation is checked. The validated and predicted outputs are directly exported to the database. The results are retrieved and displayed in the User Interface Screen according to the user input and specification.

4.4.3 Flow Chart

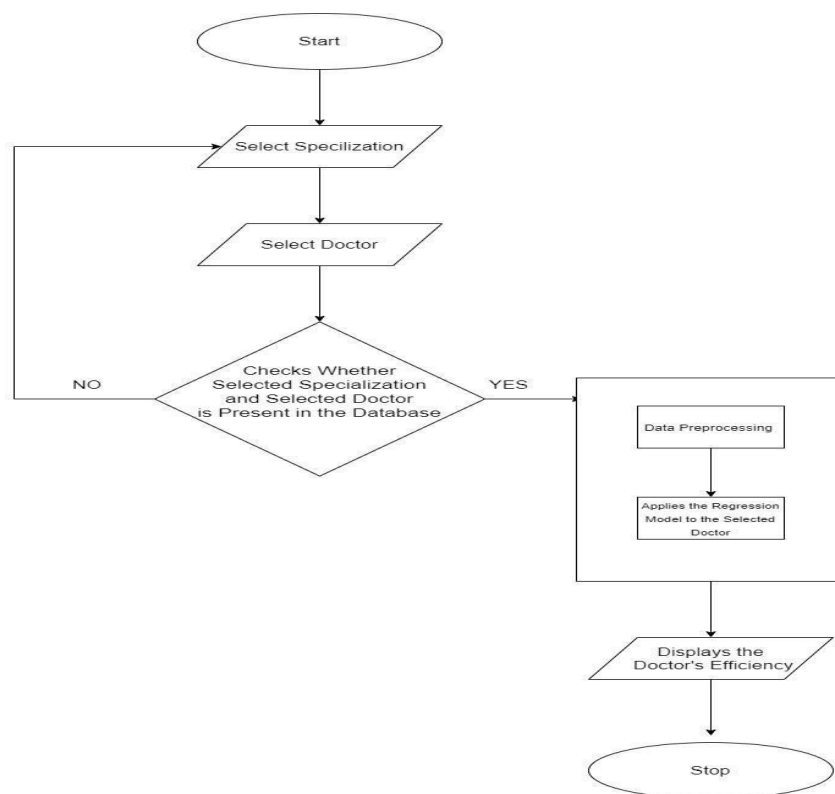


Figure 4.4.3: Flow Chart

The flow chart explains about how the flow of prediction system is done and what flow is needed to

be maintained for efficient usage of prediction system.

Flow Chart starts with UI screens where it is explained form end user i.e., hospital management side. When end user fetches for a particular doctor efficiency from 1st screen of UI as he/she need to select specialization first and select doctor to know his/her efficiency. If the efficiency is present in database then source code will run and show the output on UI screen but if not present in database recheck with the doctor name. This flow happens to all of UI screens so that dataflow does not drop in middle and no data and path failure occurs.

4.4.4 Sequence Diagram

This pictogram depicts the data flow of system between the modules. Sequence diagram also a type of UML diagram which explains about data sequence.

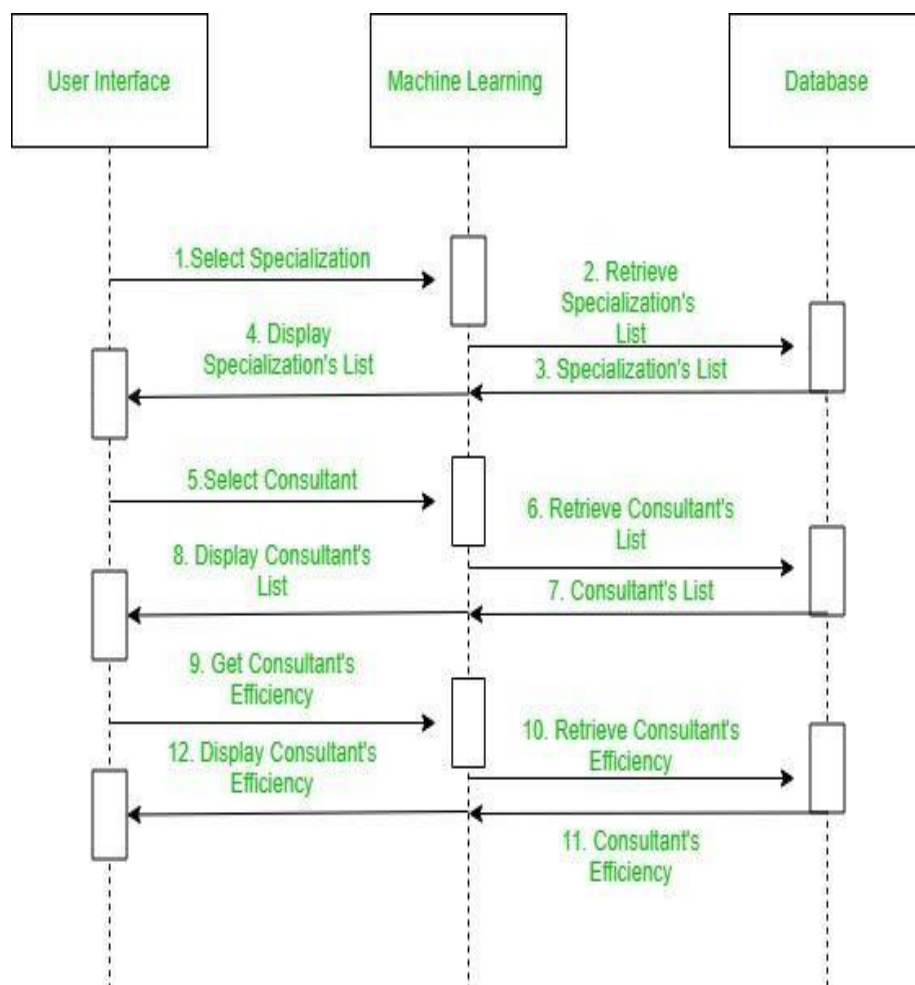


Figure 4.4.4: Sequence Diagram

The flow of instructions in different modules of data depicts a sequence. In this diagram first instruction namely selecting specialization of doctor moves from UI Screen to ML source code and from source code retrieval of specific data instruction moves to database. Then data is retrieved from database and sent to UI screen as source as middleware component where prediction system comes into play. This data sequence happens for all components of UI screen. The sequential occurrence of data is essential for a better user and developer experience.

CHAPTER 5

RESULTS AND DISCUSSIONS, PERFORMANCE ANALYSIS

The first UI screen consists of two buttons which the user will select the specialization first and doctor list will be displayed. On selecting particular doctor their efficiency is displayed.

The screenshot shows the 'Efficiency' screen of the V-PULSE Hospital Management System. The interface includes a top header with the logo 'V-PULSE' and the text 'A Hospital Management System'. Below this is a blue navigation bar with a home icon and the word 'Admin'. A left sidebar contains a menu with 'Efficiency' (highlighted), 'Find Doctor', 'Statistics', and 'Pharmacy'. The main content area features two sections: 'Specialization' with a 'Select Specialization' dropdown and a 'Doctor List' button, and 'Consultant' with a 'Select Consultant' dropdown and a 'GET ANALYSIS' button. A green box displays the text 'Dr. B.R. Mittal 'sNone Efficiency is: 87.04'. The footer contains the copyright notice 'Copyright © VNC Digital Services Pvt LTD - All Rights Reserved'.

Figure 5.1: Efficiency (Screen 1)

The second UI screen consists of a search bar where on typing a particular doctor name it displays efficiency with number of solved cases by the doctor.

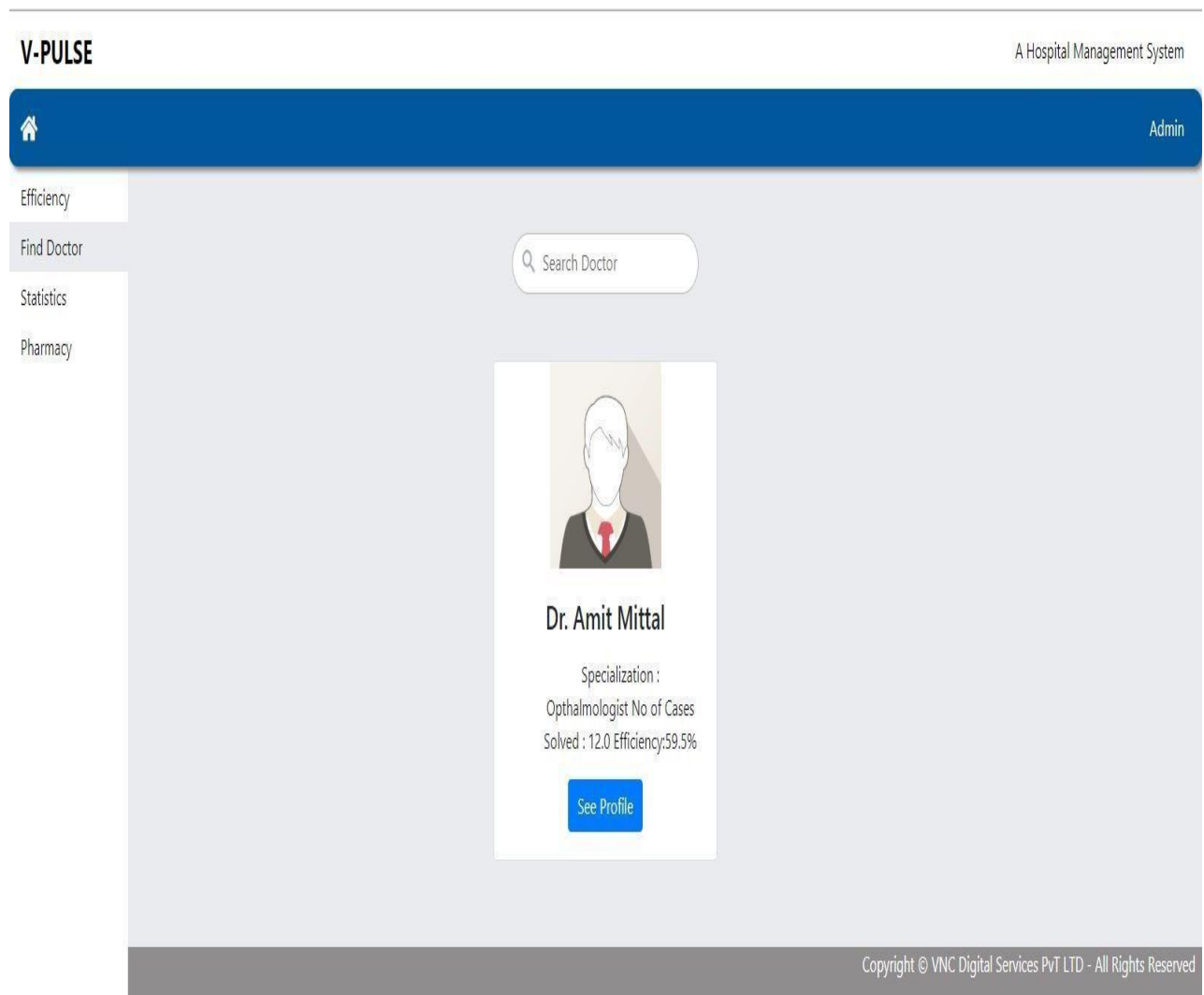


Figure 5.2: Find Doctor (Screen 2)

The third UI screen consists of a button where different specializations are listed. On selecting particular specialization top five doctors will be listed according to their rank with efficiency and solved number of cases respectively.

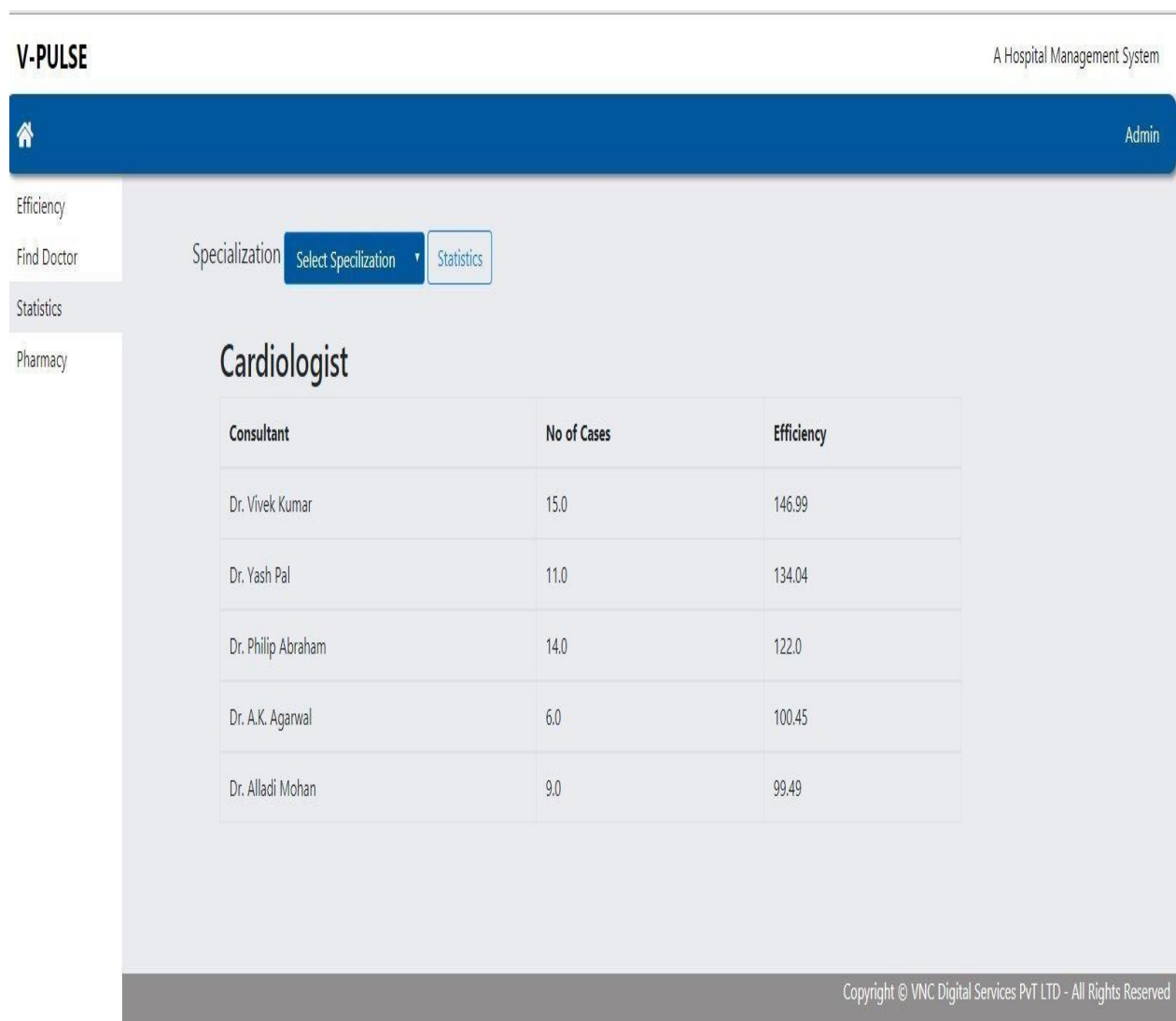


Figure 5.3: Statistics(Screen 3)

The fourth UI screen consists of a button where different departments are listed. On selecting particular department Pharmaceutical medicine list will be generated with in stock and out stock data respectively.

V-PULSE A Hospital Management System

Home Admin

Efficiency
Find Doctor
Statistics
Pharmacy

Specialization Select Specialization ▼ Stock Details

Physiotherapist

MEDICATION	IN STOCK	OUT STOCK
Corticosteroids	456	5
Theophylline	4644	515
Ibuprofen	4156	45525
Venlafaxine	5684	2465
Cholinesterase Inhibitors	11785	25

Copyright © VNC Digital Services Pvt LTD - All Rights Reserved

Figure 5.4: Pharmacy(Screen 4)

The Admin page is logged in using secured I'd and Password where they have access to add raw data to database and check the number of admin users and groups of particular users.

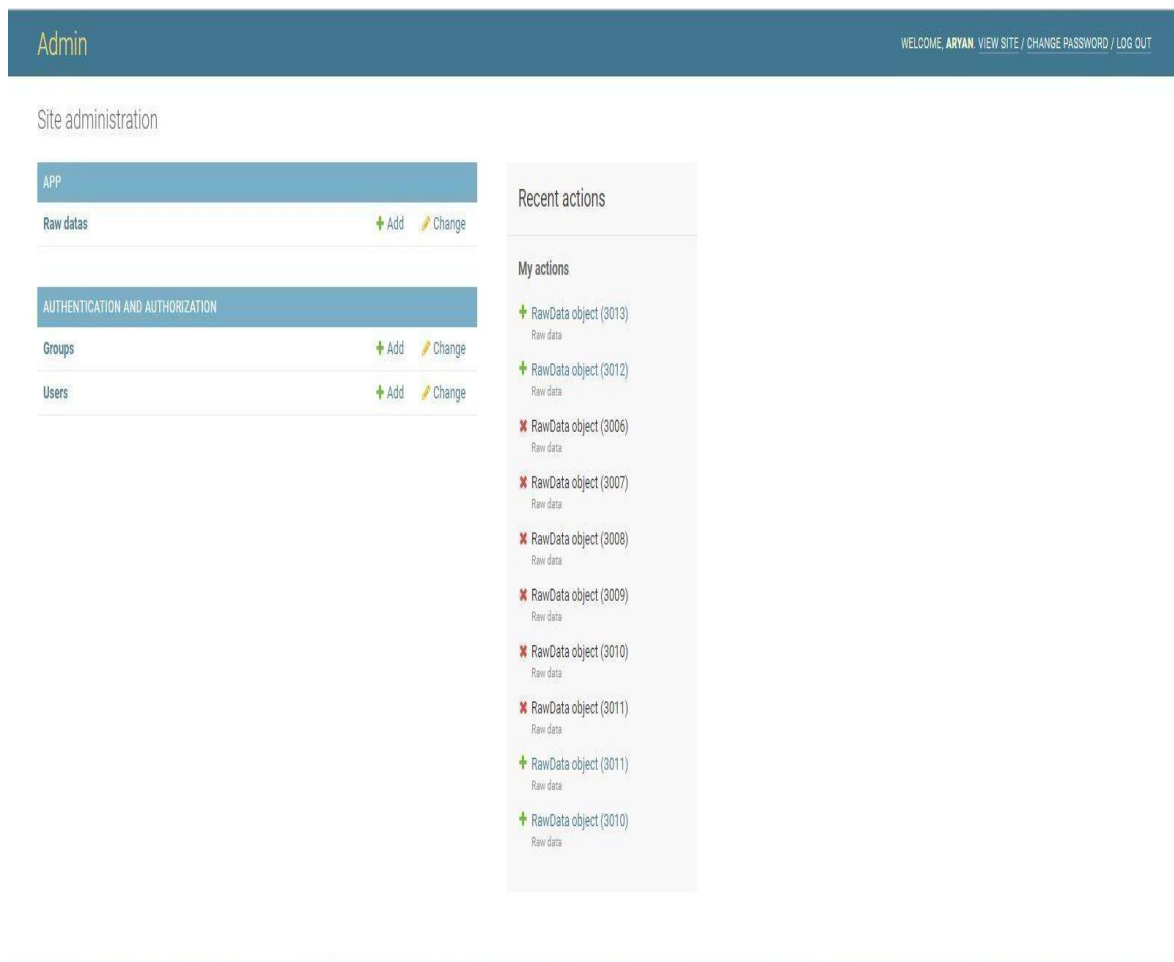


Figure 5.5: Admin page

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 CONCLUSION

This project provides a clear, feasible, easily understandable view for any HMS in knowing an efficiency of particular doctor. According to Hospital management view it is more important to now how efficient their doctors in dealing with diseases. This project helps to deal with the problem efficiently. This helps the hospital management in making better decision for future benefits of the hospital. This venture not only helps hospital management but also common people in selecting a better doctor for their problems to be dealt with. Doctor is person who plays in present society where different types of diseases are attacking people. This project uplifts some of the best doctors not only for particular organisation but for society.

6.2 FUTUREWORK

Many innovations in health care system are to make a better environment to patients as well as doctors. This project with higher accuracy in algorithm is making a clear view that prediction system in case of efficiency for doctors does not fail and Analytics 4.0 is playing a key role in predicting. This project can be updated to an app also as company is doing it. The possible enhancement is adding treatment level complexity with disease level complexity makes more clearer and system more efficient. As treatment for any disease varies upon patient resistance and symptoms he/she may get. To accomplish this task of treatment level complexity calculation Artificial Neural Networks can be used as they are used to deal with higher data in better environment. Artificial Neural Networks when deployed and trained with great variety of data for a treatment level complexity gives a better result which intensifies the Doctor Efficiency Prediction System so as it makes a great innovation in field of IT Sector. This enhancement can be commissioned under rules of VNC digital Services Pvt LTD as copy rights are taken under company norms.

REFERENCES

- [1] Zenon Chaczko, Christopher chiu, Avtar Singh Kohli, VenkateshMahadevan, 2010, "Smart Hospital Management System:AnIntegration of Enterprise Level Solutions Utilising Open Group Architecture Framework (TOGAF)", 3rd International Conference on Computer Science and Information Technology,doi:10.1109/iccsit.2010.5564121.
- [2] Pedro Luiz Cortes, Eliana Golfette de Paula C^ ortes, 2011, "Hos-^ pital information systems: a study of electronic patient records", JISTEM - Journal of Information Systems and Technology Management, vol-8 no-1.
- [3] Guo-Cheng Lan, Chao-Hui Lee, Yu-Yen Lee, Vincent S. Tseng, 2012, "Disease Risk Prediction by Mining Personalized Health Trend Patterns: A Case Study on Diabetes", Conference on Technologies and Applications of Artificial Intelligence.
- [4] Atluri Venkata Gopi Krishna, CheerlaSreevardhan, S. Karun, S.Pranava Kumar, April 2013, "NFC-based Hospital Real-time Patient Management System", International Journal of Engineering Trends and Technology (IJETT), Volume-4, Issue-4.
- [5] MIN CHEN, YIXUE HAO, KAI HWANG, LU WANG, LIN WANG, 28 June 2017, "Disease Prediction by Machine Learning Over Big Data From Healthcare Communities", SPECIAL SECTION ON HEALTHCARE BIG DATA, Digital Object Identifier11109/ACCESS.2017.2694446.
- [6] Rida Sara Khan, Asad Ali Zardar, Zeeshan Bhatti, 2018, "Artificial Intelligence based Smart Doctor using Decision Tree Algorithm", Journal of Information Communication Technology, Vol. 11 Issue. 2.
- [7] Sabyasachi Dash, Sushil Kumar Shakyawar, Mohit Sharma, Sandeep Kaushik, 2019, "Big data in healthcare: management, analysis and future prospects", Journal of Big Data, Data Object Identifier 10.1186/s40537-019-0217-0.

[8] Sunil Kumar, Maninder Singh, March 2019, "Big Data Analytics for Healthcare Industry: Impact, Applications, and Tools", BIG DATA MINING AND ANALYTICS, ISSN222096-0654 I 105/061,pp48–57Volume2,

DOCTOREFFICIENCY PREDICTIONreport

*by*PlagiarismChecker

A. PAPER

Submissiondate:27-Mar-202105:37AM(UTC-0400)

SubmissionID:1507842569

Filename:DOCTOR_EFFICIENCY_PREDICTION_FOR.edited_2_-5_1_1.docx(756.54K)

Wordcount:3829

Charactercount:21237

DOCTOR EFFICIENCY PREDICTION FOR V-PULSE APPLICATION

M MANIKANTA, V LEELA VENKATA SAI, DR. A CHRISTY

ABSTRACT- The world is composed entirely of data. Nowadays, data and information rule the world. Data analytics plays a major role in every sector of human lives. Predictive analytics, a part of data analytics, plays a vital role in different sectors of Medicine, Defense, Agriculture, Stock Exchange, and many more. Machine learning algorithms are widely used to solve many of these problems. Medicine is essential for every individual. Today, patient tracking and disease diagnosis serve as vital needs. According to the Hospital Management System (HMS), one of the major problems many undergo is doctors' recommendation to patients. Machine Learning algorithms are trained to solve this problem with higher accuracy. This can help not only management but also patients. Python Environment is used for Machine Learning Model, and Django is used for User Interface(UI) and SQLite for backend Database.

Keywords: Doctor, Django, Hospital Management System(HMS), Python, SQLite, User Interface(UI).

1.INTRODUCTION

The medical field is one of the most important and emerging sectors in today's world. Hospitals play a vital role in saving many lives and increasing global wellbeing. As a result of today's individual's mechanical lifestyle, health issues are drastically evident compared to the past. For hospitals, managing patients with upcoming health issues is a huge task to be handled effectively. Hospital Management System is the Software approach used by the majority in the medical field to maintain patient records, Pharmaceutical Records, Doctor Records, and many more. V-Pulse is an efficient application developed by VNC Digital Services Pvt LTD for HMS on the Spring Boot Platform of JAVA. Many hospitals use it for storage. It works as a Database platform and as an efficient application for Statistical analysis of

Data. HMS makes the management's work easier to maintain hospital data. Doctor efficiency is an important calculation for hospital management when considering doctors for future reference. Since patients chose their medical professionals based upon reviews and successful cases solved by a doctor or hospital management in today's world, it is necessary to predict the doctor's efficiency and rank them upon the number of cases solved and each case's complexity. The disease level complexity is defined by the previous study of many doctors' diseases and references in all aspects.

I.1 OBJECTIVE

The project's ultimate aim is to predict the doctors' efficiency based on the cases solved respectively. The existing HMS helps a patient doctor, and medical analysis but not as a prediction system. Many Doctor commendation Systems work on daily disease criteria, but our proposed system commends doctors base upon their clinicals and ranks them. The total system is developed using Python, including User Interface. The backend system is a machine learning model that uses supervised learning for clinical Prediction and Disease Level Complexity. The prediction system is a powerful efficient model where the error percentage is negligible. The data for the model is taken from the databases like MySQL, Oracle and SQLite3, depending on the hospital. The database acts as a middle for the ML model and UI model. The user interface model is designed using the Django framework in python language. There are four screens in the UI other than the admin page. The first screen shows doctor efficiency individually according to the specification list. The second screen shows the efficiency name-wise individually. The third Screen depicts the ranks of 5 doctors according to specialization. The last screen is used to check with the pharmaceutical statistics department. The Admin Page is used to manage data registration of patients, doctors, and pharmaceutical data.

1.1. LITERATURE SURVEY

The Other[1] in this paper mainly discusses the main Hospital Management System (SHS). It is an infrastructure component that relies heavily on the actual resources made available to it for its proper

functioning, operation, and maintenance. This is a solution aimed to present an architecture Integration Framework using TOGAF's Architecture Development Method. The TOGAF's Architecture consists mainly of 4 phases: Architecture Vision, Business Architecture, Information Systems Architectures, and Opportunities and Solutions, Migration Planning, Implementation Governance, and Architecture Change Management. The effectiveness of SHS is explained in terms of TOGAF with advantages and disadvantages.

The author [2] discusses the benefits and problems evident with the use of electronic versions of medical records. They constitute the permanent documentation of patient health, permitting the medical professional to evaluate symptoms and signs within a broader temporal perspective, contributing to improvements in making diagnoses and providing treatment. The hospital infrastructure systems constitute three levels: operational level, Tactical level, and Strategic level, about three functional areas of Medical areas. Administrative, and Hospitality. It also discusses the case study of testing made on HMS of three hospitals in the city of Sao Paulo, Brazil, which includes the visualization and testing results of HMS.

This author [3] discusses an effective mechanism for chronic disease risk prediction by mining the data containing historical health records and personal lifestyle information. The methodology is the traditional methodology in terms of accuracy, precision, and sensitivity for predicting the risk of diseases. In particular, the experimental results show that exploring lifestyle information can effectively enhance

whole performance for risk prediction. The framework described here is divided into three phases. In the first phase, each item's health risk patterns are found from a sequence dataset of each item in a health examination and a lifestyle dataset, respectively. Each health risk pattern is a feature attribute and a dye feature dataset for Auth Static Dynamic data in the second phase. The health risk prediction model is deployed using the classification algorithm. Finally, in the third phase, the patient's data is predicted. The **historical data related to health and lifestyle is** converted to suitable dynamic and static data according to the health risk patterns.

The author[4] explains about an eHealth patient management system based on Near Field Communication (NFC) technique. For registration, each NFC tag/wristband is identified by a Unique Identification Number (UIN) that can be programs and can also be protected through a password. NFC encoded devices can be used to read this information from the patient at any time without causing any inconvenience to them. NFC is mainly introduced to reduce cost and paperwork in the Hospital Management System (HMS). The DC-based hospital management system consists of six layers: the physical layer, middle layer, **process layer**, data access layer, application layer, and user interface layer.

The author[5] explains that **zinc** learning algorithms effectively predict chronic disease outbreaks in disease-frequent communities. A latent factor model was used to reconstruct the missing data. Convolutional neural Network- CNN-based multimodal disease risk prediction algorithm is used for structured and unstructured

data from the hospital. The prediction accuracy was about 98.4 when compared with other models in Big data analytics. The Data used was real-time hospital data stored in the cloud from the years 2013-2015, which includes 31919 patients' ZOB48 records on the whole. Stochastic Gradient algorithm and CNN-based UNIModal disease risk prediction algorithm are used for Structured and Unstructured data, respectively.

III. SYSTEM ANALYSIS

III.1 EXISTING SYSTEM

Hospital Management System (HMS) is an effective system that stores records and statistics arranged explicitly. In this corporate world, many IT companies have created many HMS applications used by many hospitals to explicitly store and analyze their data. One among the applications is V-Putse which is created by VNC Digital Services Pvt Ltd. The hospital by running hospitals has many features like the patient, doctor record maintenance, pharmaceutical data analysis with many other features. Other systems have many other features like a medical record management system, a patient treatment reminder system, and many more. Many applications have no role or function of a doctor recommendation system for management systems for patient purposes.

DISADVANTAGE:

- Many other systems do not have a prediction system or recommendation of doctors.
- The Doctor Efficiency system based on success and failure case calculation is not feasible.
- According to the Literature survey, many reliable approaches have not given a better solution to this doctor efficiency prediction system.
- Doctor recommendation system for a particular disease with the complexity of the illness is nowhere mentioned and developed.

III.2 PROPOSED SYSTEM

The proposed system is an application which recommends doctors for a particular disease with the number of successful cases solved and efficiency too. This also explores the top five doctors of particular specialization. This also has Pharmaceutical analysis according to the department wise of the concerned hospital. Our system named V-Pulsehich was developed under the Java platform of Spring Boot back-end and Angular for frontend purpose by VNC Digital Services Pvt LTD. The recommendation system was developed under Python Language. Machine Learning Algorithms are developed using the Anaconda environment in Spyder IDE. This is connected as a backend system to the middleware database. The Database is either of MySQL workbench and SQLite, which depends upon the server configuration used. The front end is the main User Interface which was developed under the Django framework in Visual Stud.

IV. SYSTEM DESIGN

The DFD depicts the flow of data in the prediction system. This flow happens between all the modules of the prediction system.

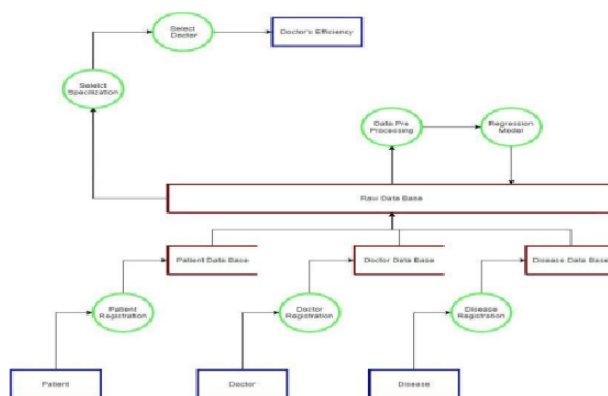


Figure 4.2: Data Flow Diagram

Primary data flow happens between the data flow of registrations. The patient details, doctor details are mainly collected and stored in a database. The disease registration happens only if a particular disease is not present in the database. The DLC classification happens using the Decision Tree Algorithm. These databases are integrated into the raw database for algorithm prediction purposes. The predicted data flow next flows back to the same database and stored in the new table. This contains values of predicted efficiencies of doctors. The next flow happens from the database to UI screens, where predicted results are displayed according to a selection of management in all four screens. The data entered in the admin page flows in the form of registrations separately in different tables and next stored in a database. This data flow happens in a transparent manner where no data loss is being happened.

IV.1. UML DIAGRAM

Unified Modelling Language diagram is used to depict the whole scenario of the model. The prediction model has different data interactions and explains them on the whole.

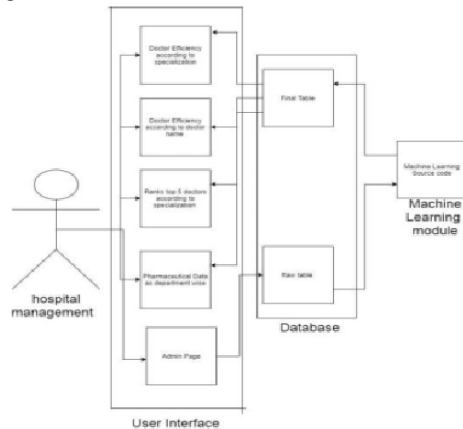


Figure 4.1: UML Diagram

They access our application directly through a web server, and details can be checked through UI screens according to their particular usage. As mentioned prior, our system has five feasible UI screens which are easily accessible and used. Data from the admin page is directly driven to the raw table and next to the source code.

In source code, the prediction system is compiled efficiently and used for efficiency prediction. The predicted efficiency data is stored in the final table. The raw table and final table are present in the same database. This data is fetched into all screens of UI as per user requirement.

The database used here is My SQL workbench and SQLite3 as depends upon the system configurations. The User Interface is developed using the Django framework of python in visual

studio. The UI screen appears in internet explorer using the static uniform resource locator of the Django framework. The data fed happens through the admin page for the patient record. The data is stored in the database and retrieved in the ML source code of the anaconda framework. The training and testing of the information happen, and validation is checked. The validated and predicted outputs are directly exported to the database. The results are retrieved and displayed in the User Interface Screen according to the user input and specification.

IV.2. ARCHITECTURE

The simpler architecture has been designed for the Prediction System. This explains all the phases of the Implementation System.

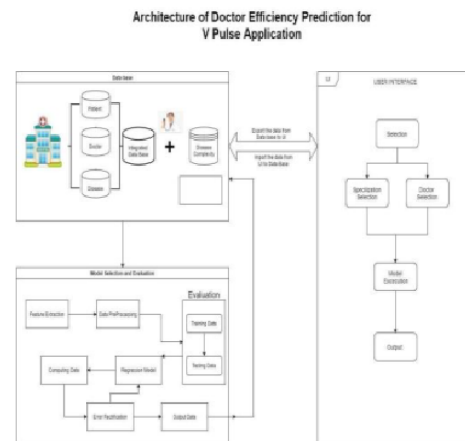


Figure 4.2: Architecture Diagram

The front end is User Interface (UI) which is connected to the middleware Database integrated with the backend Machine Learning source code. The database is a collective repository of Patient

records, Doctor Records, and Disease records. These three databases are collectively integrated into a single database with many tables. To this database, disease complexity is added from the doctor's perception, and the factors defining the disease complexity for new diseases are done through the Decision Tree Algorithm. This module is for hospital management for the Disease Level Complexity (DLC) calculation purpose. The data from the integrated database flows into the main code (or) source code.

In this source code, the initial data pre-processing happens where the noise data and irrelevant attributes are dropped. The complexity in raw data acts as a tuple. It is changed to an attribute using pre-processing techniques. Feature Selection and Feature Extraction are performed for this data for better algorithm induction. This data is split into training and testing data, and Linear Regression Algorithm is applied. The predicted data from the algorithm is compared with that of the original data and checked for accuracy. This final data is stored in the database.

This final data is stored in a new table in the database so that it is retrieved in UI screens. UI screens are created using the Django framework in python language. There are four user screens in our UI. The first screen displays doctor efficiency according to the specialization selected. One needs to select the specialization first, and the list comes accordingly. When the doctor's name is chosen, the efficiency is shown. The second screen displays the efficiency of doctors using the name of a particular doctor specifically. The third screen acts as a ranking system where it ranks the topmost

five doctors for a specific department. The final screen explores the pharmaceutical data according to the department wise. The admin page is used to check with raw data and add new data to the database. The crud operations in the database can be applied from the UI admin page itself. The number of users and groups can be checked using Admin Page itself.

IV.3.FLOW CHART

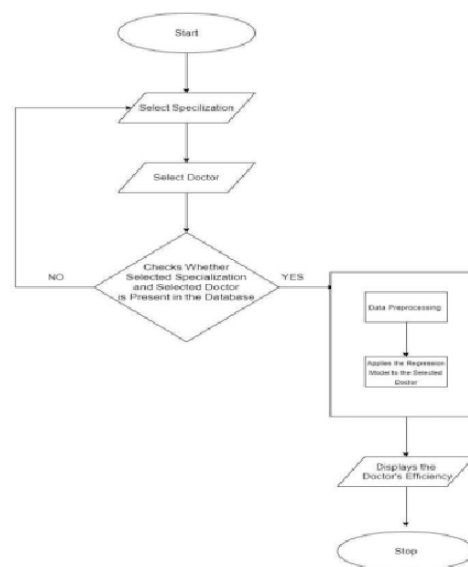


Figure 4.3: Flow Chart

The flow chart explains how the prediction system's flow is done and what flow is needed to be maintained for efficient usage of the prediction system.

Flow Chart starts with UI screens where it is explained from the end-user, i.e., hospital management side. When the end-user fetches for a particular doctor efficiency from the 1st screen of UI as he/she needs to select specialization first and the select doctor to know his/her efficiency, if the efficiency is present in the database, then the source code will run and show the output on the

UI screen, but if not present in the database, recheck with the doctor's, the name. This flow happens to all UI screens so that data flow does not drop in the middle and no data and path failure occurs.

IV.5 SEQUENCE DIAGRAM

This pictogram depicts the data flow of the system between the modules. Sequence diagram also a type of UML diagram which explains data sequence

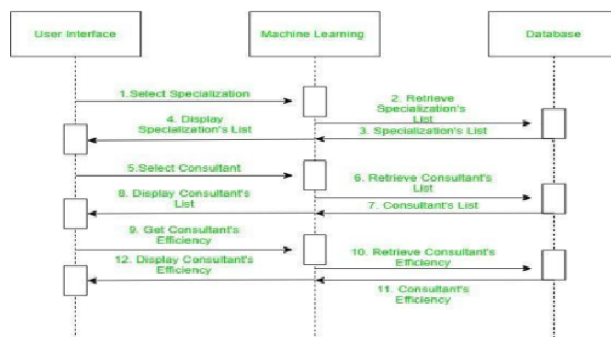


Figure 4.5: Sequence Diagram

The flow of instructions in different modules of data depicts a sequence. In this diagram first instruction, namely selecting specialization of the doctor, moves from UI Screen to ML source code, and from source code retrieval of specific data, instruction moves to the database. Then data is retrieved from the database and sent to the UI screen as a source as a middleware component where the prediction system comes into play. This data sequence happens for all components of the UI screen. The sequential occurrence of data is essential for a better user and developer experience.

IV.6 SYSTEM REQUIREMENTS

Hardware Specification

- Processor - Core i5 or i7 @1.70GHz 2.40GHz
- 10 M - 8 GB
- Hard Disk - 500 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two Button Mouse

Software Specification

Operating System - Windows 7,8,10

- ML source code environment - Python using Anaconda prompt
- DataBase - My SQL workbench or SQLite3 browser
- User Interface code - Microsoft Visual Studio for Django

Standards and Policies

Anaconda Prompt

It is a type of command-line interface that explicitly deals with the Machine Learning modules. The navigator is available in WindowsLinux and MacOS. This has a huge number of IDE's which makes it easier in coding. The User Interface also can be implemented in python.

Standard Used: ISO/IEC 27001

SQLite Database

It is a database prompt that is used to implement all SQL commands. This is used as a middleware system that contacts between the Machine Learning source code and User Interface screens. This can be integrated easily with python IDE and Django interface.

Standard Used: ISO 8601

Microsoft Visual Studio

The Microsoft Visual Studio version 16.0.12 is used for creating the User Interface. This Django framework is implemented. UI screens are designed using HTML and CSS where the framework of Django for

Integration with database and Machine Learning source code.

V. IMPLEMENTATION MODULES

MODULE 1: DISEASE LEVEL COMPLEXITY

Data Collection

Data Pre-Processing

Decision Tree Induction for Disease level

Complexity(DLC)

MODULE 2: REGRESSION MODEL

CONSTRUCTION

Data collection

Data pre-processing

Training the Regression Algorithm

Testing the Algorithm

MODULE 3: USER INTERFACE CREATION

User Interface for Doctor Efficiency according to Specialization.

Admin Page for Data Entry.

Doctor Statistics based on Specialization.

Pharmaceutical Statistics for Stock Monitoring

MODULE 4: USER INTERFACE AND

DATABASE INTEGRATION

Prediction of Doctor's efficiency

Database integration for both input raw data and output data

User Interface integration with database and scheduler for prediction algorithm.

METHODOLOGY

The Main logic of code for prediction runs in the spyder console of an anaconda environment written in python language at the backend. The database acts as middleware where all the data is stored for the process. The database used here is My SQL workbench and SQLite3 as depends upon the system configurations. The User Interface is developed using the Django framework of python in visual studio, and the UI screen appears in internet explorer using the static uniform resource locator of the Django framework. The data fed happens through the admin page for the patient record. The data is stored in the database and retrieved in the ML source code of the anaconda framework. The training and testing of the information happen, and validation is checked. The validated and predicted outputs are directly exported to the database. The results are retrieved and displayed in the User Interface Screen according to the user input and specification.

VI RESULT AND DISCUSSIONS

The first UI screen consists of two buttons in which the user will select the specialization first, and the doctor list will be displayed. On choosing a particular doctor, their efficiency is displayed

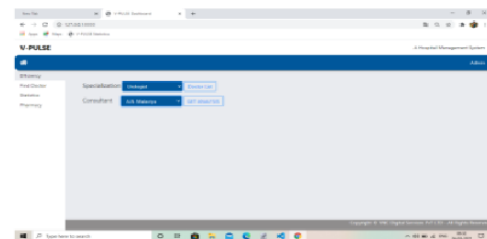


Figure 6.1: Screen 1

The second UI screen consists of a search bar where typing a particular doctor's name displays efficiency with the doctor's number of solved cases.

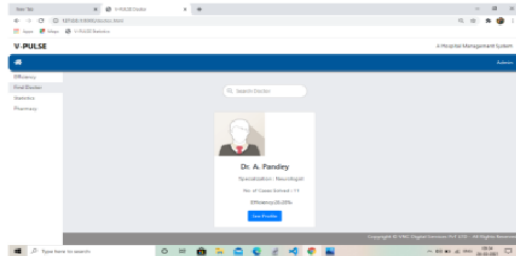


Figure 6.2: Screen 2

The third UI screen consists of a button where different specializations are listed. On selecting a particular specialization, the top five doctors will be listed according to their rank with efficiency and solved the number of cases respectively.

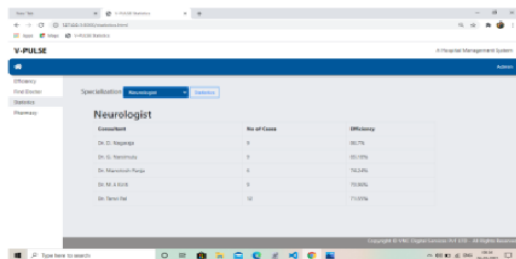


Figure 6.3: Screen 3

The fourth UI screen consists of a button where different departments are listed. A pharmaceutical medicine list will be generated with in stock and out stock data on selecting a particular department, respectively.

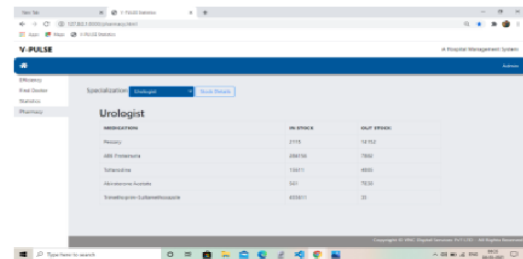
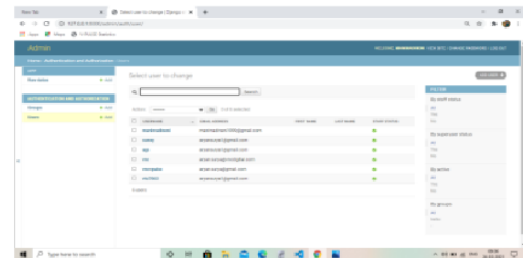


Figure 6.4: Screen 4

The Admin page is logged in using secured I'd and password, where they can add raw data to the database and check the number of admin users and groups of particular users.



VILCONCLUSION

This project provides a clear, feasible, easily understandable view for any HMS in knowing a particular doctor's efficiency. According to the Hospital management view, it is more important to understand how efficient their doctors in dealing with diseases. This project helps to deal with the problem efficiently. This allows the hospital management to make better decisions for the future benefits of the hospital. This venture helps hospital management and common people in selecting a better doctor for their problems to be dealt with. A doctor is a person who plays in a present society where different types of diseases are attacking people. This project uplifts some of

the best doctors not only for a particular organization but for society .

Future Enhancements

Many innovations in the health care system make a better environment for patients and doctors. This project with higher accuracy in the algorithm provides a clear view that the prediction system in case of doctors' efficiency does not fail, and Analytics 4.0 is playing a pivotal role in predicting. This project can be updated through an app also as the company is doing it. The possible enhancement is that adding treatment level complexity with disease-level complexity makes it more transparent and efficient. As treatment for any disease varies upon patient resistance and the symptoms, which may get. To accomplish this task, with an intermediate level of complexity, Artificial Neural Networks can be used as they are used to deal with higher data in a better environment. Artificial Neural Networks when employed and trained with a broad variety of data for a tractable level of complexity gives a better result which indicates the effectiveness of the Predictive System hence, as it makes a significant innovation in the field of the IT Sector. This enhancement can be commissioned under VNC digital services Pvt LTD rules as copyrights are taken under company norms .

REFERENCES

1. Pedro Luiz Cfirtes, Eliana Golfette de Paula Cfirtes, TO11, "Hospital Information systems: a cyclic electronic patient records", JISTEM -Journal of Information Systems and Technology Management, vol-8 no-15
2. Crux-C hung Lan. Chan- H ui , Yu-**Yen** Lee, Vincent S. **Tseng**, 2012, "Disease Risk Prediction by Mining Personalized Health Trend Patterns: A Case Study on Diabetes", Conference on Technologies and Applications of Artificial Intelligence
3. AtlurJ Venkata Crept Krishna, Cheerla Sreevardhan, S. Karun, S.Pranava Kumar, April 2013, "NEC-based Hospital Real-time Patient Management System", Trends and Technology (IfETT) , Volumes-4, Issue-4
4. MIN CHUN, Y IXUU HAO, KAI HWANG LU WANG .LIN WANG 28 June 2017, "Disease Prediction by Machine Learning Over Big Data in Healthcare Communities", SPECIAL SECTION ON HEALTHCARE BIG DATA, Digital Object Identifier 10.1109/ACCESS.2017.2694446.
5. Rida Sadath, Asad Ali Zardari, Ceshan Bhatti, 2018, "Artificial Intelligence based Sriari District Vision To Algorhythm", Journal of International Journal of Computer Science and Technology, Vol. 11 Issue . 2.
6. Sabyasachi Dash, Sushil Kumar Sliakyawar, Mohit Sharma, Sandeep **Kaushik**, 2019, "Big data in healthcare- management, analysis and prospects", Journal of Big Data, Data Object Identifier 10.1109/BigData-2019-01-01-0000000.

7. Sunil Kumar, Maninder Singh, March 2019, "Big Data Analytics for Healthcare Industry: Impact, Applications, and Tools", BIG DATA MINING AND ANALYTICS, ISSN222096-0654 1105/061 lpp48–57 Volume 2, Number 1.
8. Zenon Chaczko, Christopher Chiu, AvtarSinghKohli, VenkateshMahadevan, 2010, "Smart Hospital Management System: an integration of Enterprise Level Solutions Utilising Open Group Architecture Framework (TOGAF)", 3rd International Conference on Computer Science and Information Technology,doi:10.1109/iccsit.2010.5564121.

AUTHOR'S INFORMATION



VUKKURTHI LEELA VENKATA SAI.

Obtained bachelor's degree in computer science and engineering from Sathyabama institute of science and technology.

Specialization includes cloud computing and IOT



MADINENI MANIKANTA.

Obtained bachelor's degree in computer science and engineering from Sathyabama institute of science and technology. Specialization

includes cloud computing and IOT

B.REPORT

DOCTOREFFICIENCY PREDICTION report

ORIGINALITY REPORT

13%

SIMILARITY INDEX

8%

INTERNET SOURCES

10%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1	Guo-Cheng Lan, Chao-Hui Lee, Yu-Yen Lee, Vincent S. Tseng, Jin-Shang Wu, Chu-Yu Chin, Miin-Luen Day, Shyh-Chyi Wang, Ching-Nain Chang, Shyr-Yuan Cheng. "Disease Risk Prediction by Mining Personalized Health Trend Patterns: A Case Study on Diabetes", 2012 Conference on Technologies and Applications of Artificial Intelligence, 2012 Publication	2%
	ijettjournal.org Internet Source	2%
	www.coursehero.com Internet Source	2%
	academic.odysci.com Internet Source	1%
	Submitted to University of Greenwich Student Paper	1%
6	"New Trends in Computational Vision and Bio-inspired Computing", Springer Science and	1%

7	SubmittedtoUniversityofQueensland StudentPaper	<1%
8	repository.ipb.ac.id:8080 InternetSource	<1%
9	citeseerx.ist.psu.edu InternetSource	<1%
10	SubmittedtoUT,Dallas StudentPaper	<1%
11	"SmartCityandInformatization",SpringerScienceandBusinessMediaLLC,2019 Publication	<1%
12	SubmittedtoCampbellsvilleUniversity StudentPaper	<1%
13	MinChen,YixueHao,KaiHwang,LuWang,LinWang." DiseasePredictionbyMachineLearningOverBigDataFromHealthcareCommunities",IEEEAccess,2017 Publication	<1%
14	PedroLuizCôrtes,ElianaGolfettedePaulaCôrtes."HospitalInformationSystems:aStudyofElectronicPatientRecords",JISTEMJournalofInformationSystemsandTechnologyManagement, 2011 Publication	<1

15

Submitted to EDMC

Student Paper

<1%

16

Naira Kaieski, Cristiano André da Costa, Rodrigoda Rosa Righi, Priscila Schmidt Lora, Björn Eskofier. "Application of artificial intelligence methods in vital signs analysis of hospitalized patients: A systematic literature review", Applied Soft Computing, 2020

Publication

<1%

17

Sherwin Fernandes, Rutvij Gawas, Preston Alvares, Maeklon Femandes, Deepmala Kale, Shailendra Aswale. "Survey on Various Conversational Systems", 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020

Publication

<1%

18

V. Muneeswaran, P. Nagaraj, U. Dhannushree,

S. Ishwarya Lakshmi, R. Aishwarya,

Boganatham Sunethra. "Chapter 7 A Framework for Data Analytics-Based Healthcare Systems", Springer Science and Business Media LLC, 2021

Publication

<1%

19

wlab.cs.bilkent.edu.tr

Internet Source

20

www.uts.edu.au

Internet Source

<1%

<1%

C.SOURCE CODE

```
import sqlite3

import pandas as pd

import numpy as np

from sklearn import linear_model

from sklearn.model_selection import train_test_split

from sklearn import metrics

import schedule

import time

def run():

    conn = sqlite3.connect(r'C:\Users\Harsha
Vardhan\OneDrive\Desktop\project\db.sqlite3')

    rawdata = pd.read_sql_query('select * from app_rawdata',conn)

    #print(rawdata)

    rawcopy = rawdata

    cols=[0,3]

    rawcopy = rawcopy.drop(rawcopy.columns[cols],axis=1)

    rawcopy["count"] = rawcopy.groupby(['Consultant','Specialization']).transform('count')

    rawcopy['com']=rawcopy['Complexity']

    rawcopy.groupby(['Consultant','Specialization'])

    no_of_cases = rawcopy.groupby(['Consultant', 'Specialization'],
as_index=False).count()

    no_of_cases.drop_duplicates(subset ="Consultant", keep = False, inplace = True)

    table =

    pd.pivot_table(rawcopy,index='Consultant',columns='Complexity',values='com',aggfun
c="count").reset_index()

    table.drop_duplicates(subset ="Consultant", keep = False, inplace = True)

    #print(table)

    cols=[0,2,4]

    no_of_cases.drop(no_of_cases.columns[4],axis=1,inplace=True)
```

```

combined_data=table.join(no_of_cases)
copy_CD=combined_data
    d = {'1.0':'C1','2.0':'C2','3.0':'C3','4.0':'C4','5.0':'C5'}
copy_CD =copy_CD.rename(columns=lambda col: d.get(str(col)) if str(col) in d else
col)
copy_CD = copy_CD[['Consultant','Specialization','C1','C2','C3','C4','C5','count']]
copy_CD=copy_CD.fillna(0)
    #print(copy_CD)
copy_CD['efficiency'] =
0.2*copy_CD['C1']+0.4*copy_CD['C2']+0.6*copy_CD['C3']+0.8*copy_CD['C4']+1.0*co
py_CD['C5']
copy_CD['Total Effeciency'] = copy_CD['efficiency']/copy_CD['count']
copy_CD = copy_CD.rename(columns={'count':'cases_solved'})
copy_CD = copy_CD.drop(copy_CD.columns[8],axis=1)
    #print(copy_CD)
    comp1 = copy_CD['C1'].values
    comp2 = copy_CD['C2'].values
    comp3 = copy_CD['C3'].values
    comp4 = copy_CD['C4'].values
    comp5 = copy_CD['C5'].values
    case = copy_CD['cases_solved'].values
total_effeciency = copy_CD['Total Effeciency'].values
Cost_Len = len(comp1)
comp_eff = copy_CD[['C1','C2','C3','C4','C5','cases_solved']]
    X = comp_eff.values
    Y = copy_CD.iloc[:, 8].values
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,train_size = 0.75, test_size =
0.25, random_state = 50)
    reg = linear_model.LinearRegression()
reg.fit(X_train,Y_train)

```

```

# print(reg.intercept_)

# print(reg.coef_)

y_pred = reg.predict(X_test) ;
z_pred = reg.predict(X_train)

# print(y_pred)

# print(z_pred)

Train_comparsion = pd.DataFrame({'Actual': Y_train, 'Predicted': z_pred})
Test_comparsion = pd.DataFrame({'Actual': Y_test, 'Predicted': y_pred})

# print(Train_comparsion)

# print(Test_comparsion)

train_value = 100*(z_pred)
test_value = 100*(y_pred)

# print(train_value)

# print(test_value)

new_list = list()
new_list.append(train_value)
new_list.append(test_value)

Final_list = np.concatenate(new_list)

Final_table = pd.DataFrame({'id': copy_CD.cases_solved,
'Consultant': copy_CD.Consultant, 'Specialization': copy_CD.Specialization, 'Cases':
copy_CD.cases_solved, 'Efficiency': Final_list.round(2)})

print(Final_table)

Final_table.to_sql(name='app_results', con=conn, if_exists='replace', index=False)

train_mean = z_pred.mean()
test_mean = y_pred.mean()

# print('Mean of predicted train values:', train_mean)

# print('Mean of predicted test values:', test_mean)

train_rms_value = np.sqrt(metrics.mean_squared_error(Y_train, z_pred))
test_rms_value = np.sqrt(metrics.mean_squared_error(Y_test, y_pred))

# print('Root Mean Squared value for train data:', train_rms_value)

```

```

    #print('Root Mean Squared value for test data :', test_rms_value)
train_error_per = ((train_rms_value)/(train_mean))*100
test_error_per = ((test_rms_value)/(test_mean))*100
print('Percentage of Error for train data:' ,train_error_per,'%')
print('Percentage of Error for test data:',test_error_per,'%')
schedule.every(1).minutes.do(run)

while True:

    # Checks whether a scheduled task
    # is pending to run or not
    schedule.run_pending()
    time.sleep(0)

```